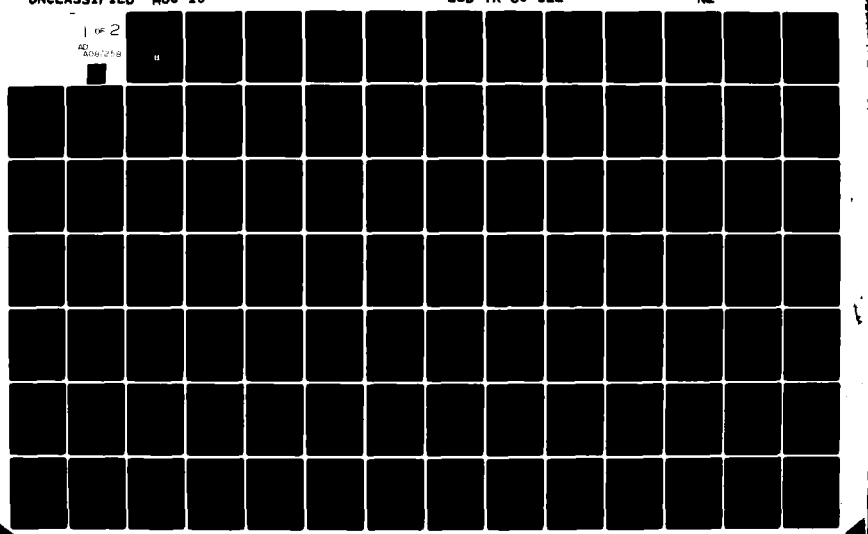
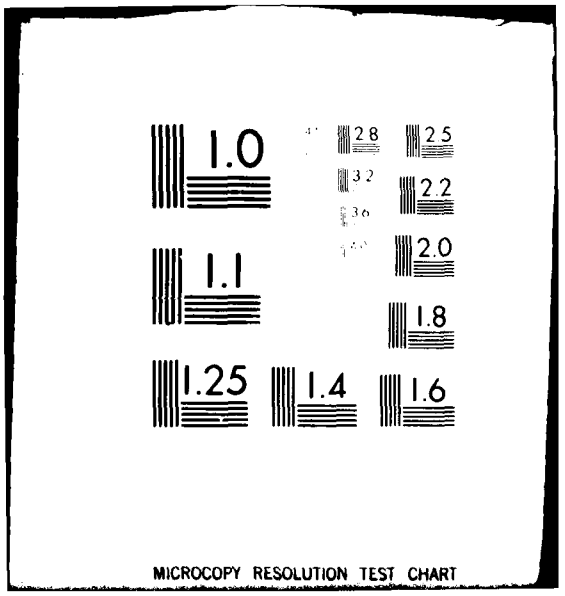
AD-A087 258 MITRE CORP BEDFORD MA F/G 17/2
REQUIREMENTS DEFINITION AND DESIGN GUIDELINES FOR MAN-MACHINE I--ETC(U)
JUN 80 S L SMITH F19628-80-C-0001
UNCLASSIFIED M80-10 ESD-TR-80-122 NL

1 OF 2
AD A087258




1.0
2.8
2.5
3.2
2.2
3.6
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

ESD-TR-80-122

LEVEL II

M80-10

REQUIREMENTS DEFINITION AND DESIGN GUIDELINES FOR MAN-MACHINE INTERFACE IN C^3 SYSTEM ACQUISITION

BY SIDNEY L. SMITH

JUNE 1980

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 572R
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-80-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

John C. Kotelly

JOHN C. KOTELLY
Computer Scientist
Technology Applications Division

Charles J. Grewe Jr.

CHARLES J. GREWE, Jr., Lt Col, USAF
Chief, Technology Applications Division

FOR THE COMMANDER

Normand Michaud

NORMAND MICHAUD, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Technical Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
(18) 1. REPORT NUMBER ESD-TR-80-122	2. GOVT ACCESSION NO. AD-A087158	3. RECIPIENT'S CATALOG NUMBER
(6) 4. TITLE (and Subtitle) REQUIREMENTS DEFINITION AND DESIGN GUIDELINES FOR MAN-MACHINE INTERFACE IN C^3 SYSTEM ACQUISITION		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER (14) M80-107
(10) 7. AUTHOR(s) Sidney L. Smith	(11) Jun 80	(15) 8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P. O. Box 208, Bedford, MA 01730	(12) 130	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 572R
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Technical Operations Electronic Systems Division, AFSC Hanscom AFB, MA 01731		12. REPORT DATE JUNE 1980
		13. NUMBER OF PAGES 119
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (1) Technical Repts.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

DESIGN GUIDELINES
MAN-MACHINE INTERFACE
MMI
REQUIREMENTS DEFINITION

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The man-machine interface (MMI) represents a significant proportion of the hardware and software investment in Air Force C^3 system acquisition. Early definition of MMI requirements and provision of design guidance in functional specification may be critical to successful system development. Tentative design guidelines are proposed in this report for further elaboration, trial application and subsequent evaluation in collaboration with selected ESD/MITRE program engineering efforts.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

The man-machine interface (MMI) is a critical element of C3 systems. There are, however, no presently available guidelines for defining MMI functional requirements and specifying MMI software design. In this report, it is recommended that ESD/MITRE should undertake a collaborative effort to explore the potential development and application of MMI design guidelines in Air Force C3 system acquisition.

The first step in MMI design is to decide what is needed. MMI requirements definition must include consideration of user/operator characteristics, the information handling requirements of people's jobs, and the functional capabilities that can be provided in the MMI. A requirements matrix is proposed, to illustrate how MMI functional capabilities can be systematically related to the demands of operator tasks.

The second step in MMI design is to develop specifications that will communicate functional requirements to the system designers. In addition to the requirements matrix, guidelines for software design should be provided with respect to dialogue type, data entry/input, data display/output, sequence control, user guidance, and other interface functions. A sample set of design guidelines for data entry functions is proposed to illustrate what might be derived on the basis of current knowledge.

Given effective requirements definition and guidelines for MMI design, an important further step in system acquisition is to ensure design review and verification before software implementation. For this purpose, specific documentation of MMI design will be needed. Such documentation will help coordinate MMI design during system development and will provide a continuing standard for any subsequent design modification. Possible approaches to MMI design documentation are proposed here for consideration.

These proposed tools for improved MMI requirements definition and design need further development and evaluation in practical application. It is recommended that a continuing effort toward that end be undertaken in cooperation with system acquisition program offices.

ACKNOWLEDGMENTS

This report has been prepared by The MITRE Corporation under Project No. 572R. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont.)

SECTION 1

INTRODUCTION

The man-machine interface (MMI) is a critical element of military command, control and communication (C3) systems. There are, however, no presently available guidelines for defining MMI functional requirements and specifying MMI software design. Steps to meet this deficiency are currently being initiated by the other military services. It is proposed that MITRE should undertake a collaborative effort with the Air Force Electronic Systems Division (ESD) to explore the potential development and application of MMI design guidelines in Air Force C3 system acquisition.

THE MMI IN SYSTEM ACQUISITION

The phrase "man-machine interface", abbreviated MMI, is frequently used by system designers, sometimes with different meanings. In the broadest sense, "man" refers to the people who will use and maintain a system. Some of these people are women. Because the operational jobs in C3 systems generally require neither exceptional strength nor dexterity, neither men nor women have any special advantage in performing those jobs.

"Machine" can refer to the tools provided people to accomplish their jobs. The tools used for C3 information processing usually include a computer with its associated terminal equipment -- displays, keyboards, printers, etc. Also important, however, are the software programs that govern the logic of computer use, the task allocation and operating procedures that give purpose and structure to a person's interaction with a computer, the paper forms which may contain data for entry into the computer, the operator manuals and other paper files which may have to be used in conjunction with computer processing, and other conditions of the work environment which influence job performance.

To acknowledge the multiplicity of factors which influence a person's use of information processing tools, the man-machine interface should be characterized in the broadest terms as person-system interaction. This is the meaning intended here: any aspect of system design that influences user performance is part of the man-machine interface.

Given this broad definition, to say that the MMI critical to C3 system design is to state the obvious. The purpose of most C3

systems is to facilitate the collection, processing and dissemination of data for human use. Thus poor design of the MMI must inevitably work against the fundamental objectives of system design. Task analysis, review of operating procedures, equipment selection, workspace configuration, and especially MMI software design -- all must be done with care to ensure effective system operation.

Not only is MMI software design critical to system operation, it can also represent a significant investment of effort in C3 system development, ranging from perhaps 10 to 50 percent of the operational software production during initial acquisition, plus software maintenance to accommodate changing operational requirements thereafter.

Given the importance and the extent of MMI software, some way must be found to determine MMI software requirements in system functional specification, to provide guidelines for MMI design, and to verify the design before implementation of MMI software. This desired sequence of requirements analysis, function specification and design verification is essentially that called for in Military Specification MIL-H-48655B (1979). What steps can system developers take to achieve this sequence in MMI software design?

NEED FOR GUIDELINES

The actual sequence of MMI software design in C3 system acquisition will sometimes depart from the principles specified in MIL-H-58655B. There may be no explicit attempt to determine MMI requirements. Specifications may include only rudimentary references to MMI design, with general statements that the system must be "easy to use". In the absence of more effective guidance, both the design and implementation of MMI software may become the responsibility of programmers unfamiliar with operational requirements. MMI software may be produced slowly, while detection and correction of design flaws occur only after system prototyping, when software changes are difficult to make.

It seems fair to characterize present methods of MMI software design as art rather than science, depending more upon judgment than systematic application of knowledge. As an art, MMI design is best practiced by experts, by specialists experienced in the human engineering of man-computer systems. But such experts are not always available to help guide system acquisition generally, and certainly cannot guide every step of MMI design at first hand. What seems needed is some way to embody expert judgment in the form of explicit procedures and guidelines for MMI design.

Present human engineering standards and design guide books are of little use to the software designer. Military standards are oriented toward hardware design ("knobs and dials") and physical safety ("avoid sharp corners"). We need similar standards for MMI software design.

MIL-STD-483 refers only briefly to MMI software design, indicating that human performance/human engineering requirements should be specified for "minimum times for human decision making, maximum time for program responses, maximum display densities of information, clarity requirements for displays, etc." (1970, page 43, paragraph 60.4.3.2.2.1)

MIL-STD-454F offers just one paragraph on the general subject of human engineering:

> Human engineering design criteria and principles shall be applied in the design of electronic equipment so as to achieve safe, reliable, and effective performance by operator, maintenance and control personnel, and to minimize personnel skill requirements and training time. MIL-H-46855 shall be utilized as a guideline in program planning and MIL-STD-1472 as a guideline in applying human engineering design criteria. Quantitative human engineering requirements shall be as specified in the contract.
>
> (1978, page 62-1)

MIL-STD-1472B (1974), the major human engineering design standard for system procurement, is concerned almost exclusively with hardware design.

Some guides to MMI software design might be reasoned by analogy. Just as we should not design sharp corners on hardware, we should not include hazardous features in user software. Just as a physical step may be marked with a painted line to keep us from tripping on it, so we may seek some way to signal abrupt shifts in the logical steps of an interactive sequence. A pushbutton reserved only for emergencies may be shielded to prevent accidental activation. So too it is possible to provide software protection in the MMI to prevent accidental initiation of critical actions. But such analogies do not take us very far.

Current efforts at re-wording human engineering standards do include token references to software, but still provide no explicit guidance, no advice on how to avoid "sharp corners" in man-computer interaction. So there is a dilemma: guidelines for MMI design seem

needed, but none are available. The question is, can needed guidelines for MMI design be developed?

CURRENT STATUS

In the past decade, as increasing experience has been gained in the use of on-line computer systems, a number of experts have attempted to set forth principles ("guidelines", "ground rules", "rules of thumb") for design of the man-computer interface. None of these sets of guidelines looks to be entirely satisfactory for MMI design.

Some guidelines have been proposed in quite general terms: "know the user" (Hansen, 1971). Some emphasize specific aspects of interface design: response time (Miller, 1968); error protection (Wasserman, 1973); command languages (Kennedy, 1974); multifunction switches (Calhoun, 1978); lightpen selection (Uber, Williams and Hisey, 1968); graphic interaction (Foley and Wallace, 1974); display formatting (Stewart, 1976; Green, 1976); color coding (Krebs, 1978).

Some guidelines have been proposed for specific operator tasks, such as document retrieval (Thompson, 1971), or process control (Williams, 1975). Some have been oriented toward the general use of on-line systems, with little task specificity (Nickerson and Pew, 1971; Palme, 1975; Chariton, 1976; Dzida, Herda and Itzfeldt, 1978). Some guidelines are based on explicit assumptions of system architecture and equipment capability (Pew and Rollins, 1975; see also Pew, Rollins and Williams, 1976). Some guidelines may assume implicit constraints, such as printed outputs rather than electronic displays (Chamberlain, 1975).

Some guidelines may be well stated generally, but are not elaborated in necessary detail (Smith, 1974). Only a few sets of guidelines have been expressed at the level of detail needed by designers. Perhaps the most detailed MMI guidelines are those proposed by Engel and Granda (1975). Here is a sample: "If a fixed length word or collection of characters is to be entered via the keyboard, limit the field on the screen by special characters, for example, underscores." More guidelines of this sort are needed.

Although this evident concern for design principles is encouraging, we still have much to learn. Most published reports dealing with the man-computer interface describe applications rather than design principles. A recent bibliography of the literature on human factors in computer systems includes 564 items, but identifies only 17 as offering design guidelines (Ramsey, Atwood and Kirshbaum, 1978). A popular current book on the design of man-computer

dialogues offers much in the way of stimulating examples, covering a range of on-line applications, but is disappointing in its failure to emphasize design principles (Martin, 1973).

Until this year, there has been no thorough-going attempt to integrate the scattered papers, articles and technical reports which have constituted the literature of man-computer interaction. A first step was made, under sponsorship of the Office of Naval Research, in compilation of the bibliography cited above. A significant follow-on effort has just culminated in publication by Ramsey and Atwood of a comprehensive summary of this literature, with particular emphasis on determining the feasibility of design guidelines.

In that compendium the authors characterize the current unsatisfactory situation:

> In some well established research areas, such as keyboard design and certain physical properties of displays, guidelines exist which are reasonably good and fairly detailed. Such guidelines may be quite helpful in the design of a console or other interface device for a system, or even in the selection of an appropriate off-the-shelf input/output device. As we progress toward the more central issues in interactive systems, such as their basic informational properties, user aids, and dialogue methods, available guidelines become sketchy and eventually nonexistent. The interactive system designer is given little human factors guidance with respect to the most basic design decisions. In fact, the areas in which existing guidelines concentrate are often not even under the control of the designer, who may have more freedom with respect to dialogue and problem-solving aids than with respect to terminal design or selection.
>
> (Ramsey and Atwood, 1979, p.2)

Summarizing their report, the authors express some reservations concerning the feasibility of general MMI design guidelines:

> Based on an extensive literature survey, this document presents a description and critical analysis of the state of the art in the area of human factors in computer systems. This review is concerned both with the status of human factors research in the area of user computer interaction and with the current state of user-computer interaction technology and practices. The primary purpose of the review is to determine whether research and

> practice in this area have evolved sufficiently to support the development of a human factors guide to computer system design. It is concluded that insufficient data exist for the development of a "quantitative reference handbook" in this area, but that a "human factors design guide" -- which discusses issues, alternatives, and methods in the context of the design process -- is both feasible and needed.
>
> (Ramsey and Atwood, 1979, abstract)

The authors do concede that useful MMI design guidelines might be developed for specific application areas, but indicate the difficulty of generalization:

> The greatest difficulty involved in making our knowledge of problem-solving aids useful to the designer is the relatively abstract level of that knowledge. If design guidelines were to be written for a very limited class of systems, it might be possible to suggest very specific aids. For more general guidelines, it would be necessary to greatly simplify the language in which problem solving has just been discussed, and provide guidance to help the designer recognize the kinds of problem-solving behavior involved in the particular tasks at hand. Furthermore, it would probably be necessary to provide very explicit examples of the various types of aids which are known. This all appears feasible, but is by no means easy, and we are unaware of any such guidelines which have been developed in the past. If designers operate primarily by synthesizing designs from already known elements, such a presentation might greatly enhance the designer's utilization of aiding techniques which happen to lie outside the designer's past experience.
>
> (Ramsey and Atwood, 1979, p.59)

Whether or not their limited optimism concerning the feasibility of design guidelines is realistic, Ramsey and Atwood have performed a considerable service by providing tabular summaries of current knowledge relating to many aspects of MMI design. A selection of their tables has been reproduced by permission as Appendix A to this present report.

A newly funded effort to develop MMI design guidelines has recently begun, under sponsorship of the U.S. Army Research Institute for the Behavioral and Social Sciences, as described in an RFP (request for proposal) titled "Development of Design Guidelines and Criteria for User/Operator Transactions with Battlefield

Automated Systems" (ARI, 1979). This study contract was awarded in November to Synectics Corporation. The Army sponsors of this effort believe that the defined restrictions on user population and information processing tasks will permit effective development of MMI design guidelines.

This current interest in MMI design on the part of Army and Navy research organizations raises the question of what is appropriate Air Force involvement. There is, of course, an element of concern. Design standards developed in one organization may eventually be imposed on others, through DoD encouragement of uniform military application. There is also an element of opportunity. In view of the Air Force's considerable experience, gained in 20 years of responsibility for the design of computer-based command systems, it seems appropriate that the Air Force should have a voice, and perhaps even play a lead role, in the development of any MMI design standards that may be established for C3 system acquisition. It is recommended that ESD/MITRE should undertake such a development effort.

PROPOSED EFFORT

There are at least three ways in which MMI design guidelines could be used. First, guidelines might help in the early definition of MMI functional requirements. This possibility is discussed in Section 2 of this report, where it is proposed to develop a matrix structure indicating the MMI requirements of various characteristic C3 information processing tasks. That approach is illustrated further in Appendix B.

Second, MMI guidelines should prove useful in the design process itself, as discussed in Section 3. The task vs interface matrix could be abstracted ("tailored") to reflect the anticipated job requirements of a particular system, and a functional description of the resulting MMI requirements embodied as guidelines in the system specification. A tentative sample of such guidelines is included in Appendix C to this report.

Third, MMI guidelines could provide criteria for the review and evaluation of a proposed interface design prior to its software implementation, as discussed in Section 4. Some means must be found for effective documentation of MMI design to permit such review. Sample formats for MMI design documentation are included in Appendix D.

To assess the potential value of MMI design guidelines, it will be necessary to try them out in actual C3 system acquisition

programs. During FY80, ESD Program Offices and MITRE engineering support groups will be invited to participate in a collaborative effort to develop and apply MMI guidelines in practice. It is hoped that several programs will volunteer to participate: programs in an early stage of system development, and anticipating a strong emphasis on the MMI in system design. Ideally, programs will be selected which include a wide variety of different operator jobs, in order to encourage a broad consideration of applicable MMI guidelines.

Whatever MMI guidelines are developed at this stage can only be regarded as tentative, to be used on an interim basis until experience is gained in their trial application. The initial development process may have to extend over a period of several years, since it is necessarily linked to the pace of system acquisition. Such an extended effort may well be justified, however, in view of the potential long-term benefits. In the long run, standards for MMI design established at ESD/MITRE could contribute significantly to improved C3 system acquisition.

David Penniman, writing for the User On-Line Interaction Group of the American Society for Information Sciences, cites the same need for "An interim set of guidelines for user interface design based on available literature and pending the development of better guidelines as our knowledge increases" (1979, page 2). Penniman goes on to remind us that interim guidelines are better than no guidelines at all.

There is one important factor to consider in this regard, which is the relative stability of human engineering guidelines. Standards for hardware design may change as each new generation of equipment becomes available. But people change hardly at all from one generation to the next, in terms of their basic information processing capabilities and limitations. This relative invariance of people with respect to technology poses both problems and advantages.

The most significant advantage is this: if MMI guidelines can be expressed in terms of basic human capabilities rather than transient technology, a design standard of enduring value will be established. It may be true that such guidelines can be established only slowly, but that is all the more reason to make a start now.

SECTION 2

MMI REQUIREMENTS DEFINITION

The first step in MMI design is to decide what is needed. Once MMI requirements are determined, then design guidelines can be tailored to anticipated needs. MMI requirements definition must include consideration of user/operator characteristics, the information handling requirements of people's jobs, and the functional capabilities of the MMI that are needed for those people performing those jobs.

USER/OPERATOR CHARACTERISTICS

A challenging aspect of C3 system design is the broad range of user characteristics that must be accommodated. The users who operate C3 systems include a wide variety of people with different degrees of training and experience. The skill mix may include command managers as well as clerical personnel, hardware and software technicians performing maintenance functions, even "box kickers" at a truck dock in the automation of air cargo data handling (Smith, 1976b).

There are various ways to categorize the different kinds of users. One categorization distinguishes "operators" and "analysts" from "service" personnel (Goodwin, 1978; see also Rouse, 1975). In this view, an operator is a person performing a structured task, usually at a forced pace, monitoring and controlling through system outputs and inputs, making decisions and initiating actions through the system within limited time constraints: e.g., radar track monitoring; air traffic control; process control. An analyst performs an unstructured task, usually self-paced, manipulates system data to establish relationships, perhaps using on-line tools, may rely also on data from other sources including past experience to recognize problems and make decisions, which may not be made immediately and may not be entered into the data processing system for action: e.g., intelligence analysis; mission planning; message handling. Service personnel perform more routine clerical tasks, where the user is not the source of data being entered, retrieval is in response to specific query, using highly structured transaction sequences, sometimes at a forced pace under pressure: e.g., making airline reservations; providing telephone directory assistance; word processing jobs.

Although such categorization offers helpful descriptive information of the user's role, it is clear that the characterization is largely one of differences among jobs rather than differences among people. Job differences can be defined better in the more specific context of task analysis. For purposes of the present discussion, no attempt will be made to label user differences in terms of job description. The terms "user" and "operator" will be used interchangeably throughout this report.

For any one category of user, individual differences in skill may be considerable. Because of systematic rotation of job assignments for military personnel, today's naive user becomes next year's expert, then to be replaced by another beginner. This regular reassignment policy implies the need for flexible job aids in MMI design, which can provide optional help to the novice user but can be bypassed by the expert.

Where there is high personnel turnover, low skill levels and minimal operator training, MMI design must take that into account. Interactive sequences should be configured as a routine series of simple steps, with adequate guidance and on-line error correction procedures to ensure effective operator performance. For most C3 applications, however, both operator characteristics and job demands will require an MMI design offering more extensive capabilities and permitting more flexible use.

An important distinction can be made between dedicated and casual users (Martin, 1973). C3 systems include both kinds. An operator in a surveillance job may spend virtually full time monitoring computer-generated displays. Such a person will generally receive extensive training, and the MMI design can be tailored to his presumed skills. Another operator may be an intelligence analyst who interacts with a computer system only periodically, and whose requirements for information retrieval are much more wide ranging. Although this person may have been instructed in MMI procedures, perhaps being given an operator's manual, his use of computer aids may remain uncertain for an extended period unless on-line guidance is incorporated in MMI design.

Although these examples are cited to illustrate user characteristics, they actually reflect important differences in job requirements. It is obvious that user characteristics are closely related to job requirements, and that necessary user skills can be specified more exactly after job requirements have been determined. This is discussed further below.

As a general objective, it can be argued that the MMI design in automated systems should not require higher skills and greater operator effort than the manual procedures which are being replaced. In practice, of course, this design objective is not always attained. Some automated systems are found to be harder to use rather than easier, and increased demands are made upon the operators.

In the Army RFP cited earlier, the increasing demand for skilled operators caused by command automation is advanced as an argument in favor of developing MMI design guidelines:

> The skill/demand mismatch is due in part to the fact that the equipment/procedural configurations of existing and projected systems have been devised without coordination among proponents of different systems. As a result, very little of the skill and know-how accrued from experience with one system can be transferred to other systems. Therefore, one of the long range goals of this project is to promote functional standardization and modularization of user/operator tasks and procedures in order to reduce the amount of training and the skill levels required of users/operators of fielded automated data processsing systems. Many of the procedures employed in the operation of various automated systems are basically similar. Yet from the user/operator's perspective each system is a new situation with little carryover or transfer from previous exposures to other systems. While it may be too early to establish absolute "standards" for many system parameters, a measure of consistency would go a long way toward increasing effectiveness, reducing personnel costs and making battlefield automated systems more approachable to Army users/operators.
>
> (ARI, 1979)

This call for consistency in MMI design is common to all recommendations on the subject, but is particularly important in military settings because of the frequent rotation of personnel. To the extent that jobs may be similar in different C3 system applications, the operator transferred from one system to another ought to be able to use similar means to accomplish similar ends. Even for quite different system applications, it may prove possible to introduce a coherent general approach to MMI design, so that the interface logic will seem familiar to a person transferring to a new job.

For the developers and designers of C3 systems, it may help to postulate some basic characteristics of the prospective users. The operators of C3 systems will generally be intelligent men and women, with professional military skills. These people will not necessarily be knowledgeable about computer technology, may have little time to learn specialized interface procedures, and will have various degrees of familiarity with the system. Being human, these operators will sometimes make mistakes, especially when working under pressure, and good MMI design must take this into account.

These operators are motivated toward effective job performance in the face of operational demands. They will generally regard automated data processing as a tool to aid job performance, with little curiosity about the internal mechanisms of computing machines. Operators will tend to judge the entire system on the basis of their personal experience with the MMI. If the MMI is efficient and easy to use, operators will like the system. But users will be impatient and critical when handicapped by a clumsy interface design.

For further information on user characteristics, see Tables A-1 through A-3 in Appendix A.

TASK ANALYSIS

Fundamental to MMI design is the analysis of operator jobs. The process must begin with the mission requirements of a proposed C3 system, which state the basic objectives to be accomplished. These mission requirements are then elaborated and translated, taking into account the proposed operational employment concept, environmental, technological and fiscal constraints, to define the system operational requirements.

Operational requirements imply the performance of various identifiable functions -- data sensing, data transmission, data processing, etc. Analysis of those functions, in turn, establishes more specific data processing requirements -- what data must enter the system, what data must be stored, what combinations and transformations of data are required, what kinds of information should result from that processing.

These data processing functions imply the specification of tasks to accomplish particular ends. Some tasks may be performed entirely by machine and thus affect MMI design only indirectly if at all. Because of the critical role of human judgment, however, many tasks in C3 systems involve joint performance by man and machine.

Most tasks can be partitioned into identifiable subtasks. Those subtasks in turn are often designed as a series of simple, discrete transactions, such as operator entry of a single item of data. (As defined here, note that a transaction is the smallest functional "molecule" of man-machine interaction, and does not denote an extended task sequence.) It is at these levels of task, subtask and transaction that MMI design guidelines might be applied.

What about jobs, where do they fit in? To define a job, which is a set of responsibilities and activities for a person, we must allocate tasks to operators. Task allocation, however, will not prove easy. For C3 systems it will seldom be desirable to allocate this task to man and that task to machine, as textbooks suggest. Instead, more effective performance may result when man and machine participate jointly. This view has already been stated above, and is emphasized also by Ramsey and Atwood (1979).

If this is the case, then a more accurate description of job definition involves specification of operator participation in tasks. From this viewpoint, one possible contribution of MMI guidelines would be to suggest the most effective ways in which operators can participate in various identified C3 information handling tasks, as well as to specify the functional MMI capabilities needed to facilitate such participation.

It should be noted that jobs as defined here are generally not the same as information processing functions. Jobs can be both smaller and larger than functions. A surveillance function, for example, might involve a monitoring task performed by several different operators, perhaps each responsible for a different surveillance sector. And each of those operators, as part of his job, might also participate in other tasks related to different functions, for example, weapons control.

Thus job definition, the combining of tasks performed by people, involves something more than functional analysis, an extra application of judgment in the design process. The purpose of job definition is twofold. On the one hand, the definition of jobs determines what capabilities will be required of the operators. On the other, the definition of jobs indicates what tasks may have to be performed in combination, by a particular operator at a particular work station, and so implies what functional capabilities will be required of the MMI.

FUNCTIONAL CAPABILITIES

If an operator must interact with a graphic display, as in a computer-aided design task, he will need a functional capability for pointing at different parts of the display. Without a pointer to designate displayed objects and positions, the task would be difficult or even impossible to perform. Various means might be chosen to provide a pointer -- a lightpen, perhaps, or joystick-controlled cursor, or trackball, or whatever -- each choice offering its own advantages and disadvantages. But it is the functional requirement for pointing that is essential.

For an on-line editing task, where an operator must designate arbitrary portions of displayed text for correction, a pointer would clearly be useful. For a task involving sequential selections among computer-displayed options, a pointer would probably be useful, although other design alternatives such as multifunction keys might do nearly as well. For many other tasks a pointer may not be needed at all.

One could imagine making a long list of typical C3 data handling tasks and noting for each one whether a pointing capability is an essential or probably useful feature of the MMI, or whether it is not needed. To do this would be to make a modest beginning in the definition of MMI requirements. But, of course, many other MMI capabilities have to be considered also.

How about color? Color coding could be very helpful to an operator monitoring a complex display, trying to detect critical conditions in rapidly changing data, but for many other tasks the expense of colored displays might not be justified. Perhaps blink coding, special symbols or auxiliary alarms could be used instead.

Many functional capabilities of the MMI have little to do with hardware selection, and depend primarily on the logic of software implementation. Consider two examples. For a particular data entry task, an operator may need a capability to defer entry of unavailable items, even when they may be essential for subsequent data processing calculations. For many data entry tasks operators may need a flexible capability to back up and change previously entered items.

The software designs chosen to implement these two functional capabilities -- deferred data entry and optional backup -- might share some common features, e.g., some sort of suspense file. Software implementations would also differ in some respects, e.g., in terms of the validation checks applied during data entry, those

applied at later stages of data processing, and the particular messages displayed to the operator when data deficiencies are detected.

It is safe to say that the operator will not care what goes on behind the scenes, but will simply use the operational capabilities available to him, however they are provided. And it is these capabilities that should be described in functional MMI specification, rather than their means of implementation.

As more and more MMI capabilities are considered, our initial list of which tasks require a pointer will expand into a sizable table, or "matrix" as it will be called here. Across the top, labeling each column, will be the names of tasks. Down the side, labeling each row, will be listed the various different functional capabilities we have considered. In the body of the matrix, at the intersection of each column and row, there could be a cell entry indicating whether that row/function is essential, useful or not needed for the performance of that column/task.

There are several problems with such an MMI requirements matrix. The chief problem is that it does not yet exist. No one has yet undertaken the systematic effort needed to create it. Laboratory researchers have been too removed from operational applications to see the need for such a matrix. System developers, on the other hand, have enough trouble defining the MMI features required by a particular application, let alone trying to assess the broader range of capabilities which might be required by other tasks.

What may be needed here is a broad survey across systems, of the kind proposed in the Army RFP cited earlier. Pending such a comprehensive effort, a beginning might be made by tabulating several characteristic tasks and interface features, perhaps starting with one C3 system and then moving on to another, with the intention of expanding the MMI requirements matrix with increasing experience in its use. Such a beginning with a rudimentary matrix is illustrated in Appendix B to this report. See also Tables A-4 through A-9 in Appendix A.

Other problems associated with the MMI requirements matrix, aside from its present unavailability, have to do with the labeling of rows, columns and cell entries. As suggested above, rows should be labeled as functional capabilities. This ideal may be difficult to attain when our thinking turns so easily to implementation. Current lists of desirable MMI features tend to include means (e.g., lightpen) more often than ends (pointing). As a practical matter,

we shall have to begin with whatever comprehensive lists we can find, and try to abstract the implied functional capabilities.

Column/task labeling poses similar problems. If a column represents a task defined specifically for a particular system, perhaps monitoring a track display for air traffic control, then the resulting pattern of MMI requirements may not provide useful guidance in specifying a somewhat different but related task in another system, perhaps monitoring a track display for air defense purposes. To prevent the MMI requirements matrix from growing indefinitely, as each new system acquisition adds its own new set of specific column/tasks, it will be necessary to generalize task descriptions to some degree, to list characteristic rather than specific C3 tasks.

With further investigation, it is possible that we shall find the key to successful generalization at the level of subtasks rather than tasks. Consider the obvious subtask of keyed data entry, which is a common component in many larger tasks. (The expressed commitment to hardware implementation in the word "keyed" might be forgiven in view of currently accepted limitations in interface technology.) MMI requirements determined for that subtask should prove applicable in a variety of C3 system applications, with perhaps some occasional modifications appropriate to special working conditions.

Or it may be that to derive generic MMI requirements one must distinguish several categories of a subtask, such as keyed data entry with prompting or without. Whatever the level of task specification that eventually proves useful, there seems the potential here for using some form of abstracted or generic task analysis to predict the MMI requirements of actual tasks. In future, when our understanding of generic task analysis has increased, we might look at a proposed operator task, estimate it as "60 percent monitoring a high-density multivariate display, 15 percent controlled option selection, 25 percent prompted keyed data entry", and derive from that description a specification of MMI functional requirements.

Today we do not know how to do that. But as we add column/tasks to the MMI requirements matrix we should stay alert to the possibilities for generalization, trying to abstract task descriptions wherever possible and to identify the subtasks and their variations which may comprise the generic components of future requirements definition.

Aside from the problems of selecting suitable row and column labels, we also have to depend largely on judgment in designating cell entries for the MMI requirements matrix. At present, probably the best we can do is make an approximate categorical judgment that MMI capabilities are essential, useful, or not needed. In future we may be able to replace those approximate estimates with more quantitative weighted values. It might be possible to derive quantitative values from system performance measures, although that seems unlikely at present. More probably, we could develop quantitative values directly from structured testing of generic task components, as those can be identified.

Still another problem with the MMI requirements matrix is that rather different versions of the matrix might conceivably be needed for different categories of users. This problem, however, may be more conceptual than real. As a practical matter, several arguments can be advanced in favor of a single matrix. As noted earlier, user characteristics are implicit in task descriptions: simple clerical tasks are often assigned to untrained personnel, complicated clerical tasks to experienced, well-trained people, monitoring and control tasks to young officers, planning and resource allocation tasks to older officers, etc. Moreover, because of the skill mix of operators in most C3 jobs, MMI functional specifications for any particular job must generally accommodate a fairly broad range of user characteristics.

In the future, when the MMI requirements matrix may be expressed in terms of generic task components, the results of functional analysis may be expressed in terms of required operator skills, as well as specification of MMI requirements. Thus again a single requirements matrix may be referenced for more than one category of system user.

Nonetheless, it will prove a wise precaution to consider user characteristics explicitly when applying the MMI requirements matrix. A prospective user group may be "special" in some significant way, as might be the case in a C3 system designed for operators in another country with different linguistic background and cultural expectations. A user group may include color-blind people, or people with some other special characteristic. If so, then those special user characteristics should be taken into account and MMI requirements interpreted accordingly.

In the face of these various problems, why should we attempt to develop an MMI requirements matrix? What advantages will it offer? First, it seems clear that such a matrix should help to provide general perspective and serve as a framework to structure judgment

in determining MMI requirements. This would be true even in the early, rudimentary stages of MMI requirements matrix development.

With further development, as the MMI requirements matrix expands to include more functional capabilities and a manifold range of C3 tasks, it will embody the experience accumulated in a variety of system acquisition programs. It may then be possible to draw on this cumulative judgment instead of having to re-invent an MMI from scratch for each new system. If nothing more, the row labels in the matrix would constitute a checklist to remind system developers of MMI capabilities that should be considered.

In the long run, of course, the MMI requirements matrix could become an even more powerful tool. As in the hypothetical example described earlier, it may eventually be possible to analyze prospective operator tasks into familiar subtasks and their generic components, for which MMI functional requirements have already been determined. If so, then requirements definition for a new system could be accomplished both rapidly and comprehensively.

An important use of functional requirements definition in early stages of system acquisition will be to limit the choice of MMI design guidelines to be employed during system development. As discussed later in this report, the total "catalog" of design guidelines might grow quite large, perhaps including hundreds of items. But many of these guidelines will be specifically related to particular functional capabilities. Thus when the subset of capabilities required by a new system has been determined, then a related subset of guidelines could be selected, tailored to system requirements.

One could imagine providing computer aids to facilitate this tailoring process. If both the MMI requirements matrix and the design guidelines were presented in computer storage, with appropriate cross indexing, then specification of desired functional capabilities could be programmed to produce automatic printout of the corresponding set of guidelines for MMI design.

By its nature, the MMI requirements matrix cannot be created all at once, but will grow gradually through accretion, as new capabilities and tasks are added and old ones are further analyzed and subdivided. After several years of experience using the matrix, some estimate might be made of the eventual success of this approach to MMI requirements definition. To gain that experience, a beginning must be made in current C3 system acquisition programs, as a collaborative effort with program personnel. Such an effort is recommended.

SECTION 3

MMI DESIGN GUIDELINES

The second step in MMI design is to develop specifications that will communicate functional requirements to the system designers. Specifications may include descriptions of the operational work environment and interface hardware, if those are expected to constrain MMI design, but will provide critical guidance for the design of MMI software with respect to the selection of dialogue type, and the requirements for data entry/input, data display/output, sequence control and user guidance.

WORK ENVIRONMENT

In itself, the design of a work environment seldom fulfills functional requirements directly. (Exceptions might be noted for the special requirements of merchandising, stagecraft, etc.) Instead, the general objective of workspace design is to minimize constraints on functional performance.

In many C3 system applications, the immediate work environment is relatively benign, and does not constrain MMI design. But there are exceptions. For example, the noise, dirt and confusion of movement at a truck dock may limit MMI options available for air cargo data entry (Smith 1976a; 1976b).

Even in clean, quiet office locations, poor workspace layout can handicap operator performance. Inaccessibility of paper files, inadequate workspace at an operator's station, badly positioned displays and keyboards, glare sources in unbalanced ambient illumination, all take their toll. It can be argued that environmental deficiencies such as these contribute in large measure to complaints of operator fatigue after long hours at a constrained work station.

Although an optimal work environment can sometimes be specified in C3 systems, when that is not possible some limits on MMI design may have to be imposed in order to accommodate sub-optimal working conditions. As an example, mechanical vibration in airborne command posts might hinder fine manipulations required for lightpen use. A noisy environment can obscure voice output displays and other auditory signals. A high ambient illumination, such as experienced in an open control tower on a sunny day, can reduce the

effectiveness of light signals and other visual displays, which may have to be taken into account for some system applications.

Where a normal working environment can be assumed, it may be possible to rely on standard procurement specifications and engineering practice to produce a good design. Where unusual working conditions must be accommodated, however, it will be necessary to include a description of environmental constraints in the MMI design specification.

INTERFACE HARDWARE

Although it is usually the logic and software implementation of MMI design which prove most critical, hardware choices can affect implementation and eventual system performance. Hardware here is meant to include input devices, output display and signaling devices, and also printouts, paper forms and other equipment which may be used in conjunction with the MMI. If the MMI is used to mediate person-to-person communication, which is becoming more common, then communication facilities should probably also be included under this heading.

Functional specifications for hardware, such as those for display capacity, legibility, etc., are reasonably well understood in C3 system acquisition and will not be considered further in this report. Such device capabilities should be included in an MMI functional specification, however, if the physical requirements can be determined in advance of MMI design. Ideally, of course, hardware specification should follow and derive from functional design. In practice, it may happen that C3 system acquisition may build upon and hence be constrained by existing equipment.

The effect of hardware constraints on MMI design may be more subtle than suggested by physical equipment specifications. As an example, the technique and format of data input may have to accommodate the nature of the data source, paper forms being transcribed, etc. For data output, display formats may have to be adaptable to document printing and other constraints. For sequence control, hardware choices can have a pervasive influence. As an obvious example, the availability of a lightpen or other pointing device for selection of control options will permit more flexible display formatting than the use of multi-function keys labeled in display margins.

Where anticipated hardware limitations will constrain MMI functional capabilities, these must be indicated clearly in the design specification, so that they can be taken into account and the

functional design compromised as necessary. Where MMI hardware can be chosen freely, as in acquisition of a new system, it may be desirable not to impose any special constraints, except those implied in equipment design standards. The inclusion of detailed equipment "requirements" in system specification may be thought to substitute adequately for the lack of detailed functional specifications. For MMI design, however, with its heavy dependence on software for effective implementation, detailed hardware specification may make good design harder to achieve rather than easier.

DIALOGUE TYPE

A fundamental decision in MMI design is selection of dialogue type(s). Here dialogue refers to the sequence of transactions which mediate man-machine interaction. Ramsey and Atwood (1979, pp. 76-95) identify eight general dialogues:

question-and-answer
form-filling
menu selection
function keys with command language
user-initiated command language
query languages
natural-language dialogue
interactive graphics

Various sub-categories can, of course, be identified, as illustrated by the many examples in Martin's book on the subject (1973).

In C3 systems, MMI design will often involve a mixture of two or more dialogue types, since different dialogues are appropriate to different jobs and different kinds of users. Recognition of appropriate dialogue types at the outset of system development will facilitate MMI design and help ensure the effectiveness of eventual system operation.

The selection of dialogue type based on anticipated task requirements and personnel skills seems straightforward, at least for simple cases. Computer-initiated question-and-answer dialogues are suited to routine data entry tasks where data items are known and their ordering can be constrained, and provides explicit prompting for unskilled, occasional operators. Form-filling dialogues permit somewhat greater flexibility in data entry, but may require operator training. When data entries must be made in arbitrary order, perhaps mixed with queries as in making airline

reservations, for example, then some mixture of function keys and coded command language will be required for effective operation, implying a moderate to high level of operator training.

From these examples, it would seem possible to judge for any information handling task the type(s) of dialogue that should prove most suitable, and to include this judgment in the MMI requirements matrix and possibly in the functional specification as well.

One important aspect of dialogue selection is that different types of dialogue imply differences in system response time for effective operation. In a repetitive form-filling dialogue, for example, the operator may accept relatively slow computer processing of a completed form. If the computer should take several seconds to respond, the operator probably can use that interval to set one data sheet aside and ready another. But several seconds delay in a menu selection dialogue may prove intolerable, especially where the operator must make an extended sequence of selections in order to accomplish a desired end.

Table 3-1

Estimated Requirements of Different Dialogue Types

Dialogue Type	Required User Training	Required System Response Time
Question and Answer	Little/None	Moderate
Form Filling	Moderate/Little	Slow
Menu Selection	Little/None	Very Fast
Function Keys with Command Language	High/Moderate	Fast
User-Initiated Command Language	High	Fast
Query Languages	High/Moderate	Moderate
Natural-Language Dialogues	Moderate (potentially Little)	Fast
Interactive Graphics	High	Very Fast

To categorize these differences, Table 3-1 presents for each of the eight general dialogue types identified above an estimate of the implied requirement for operator training and for system response time. Cumulative experience and specific requirements of a particular task may lead us to modify such estimates. But the general principle illustrated here, that one design choice implies others, must be taken into account in MMI specification.

Specification of dialogue type will also determine other aspects of MMI design. Guidelines specifically appropriate to form-filling dialogues may be irrelevant for the design of other kinds of dialogue. Eventually it may be desirable to code guidelines in some way, to designate those pertinent for each type of dialogue. Such coding would facilitate the tailoring of guidelines to meet the MMI requirements of particular tasks.

For further comment on dialogues and system response time, see Tables A-10 through A-13 in Appendix A.

DATA ENTRY/INPUT

Another way to categorize design guidelines is in terms of their relation to general MMI functions. A fourfold distinction is adopted here, to permit discussion of guidelines relating to data entry, data display, sequence control and user guidance. Data entry refers to input by the operator of data items to be processed; command inputs or option selections intended to control data processing are considered later in the discussion of sequence control.

Data entry is heavily emphasized in tasks related to clerical jobs, and many other C3 tasks involve data entry to some degree. Because data entry is so common, because the requirements of data entry seem to be readily understood, and because inefficiencies caused by poorly designed data entry are so apparent, many of the published recommendations for good MMI design deal with this topic.

Rather than trying to present here a long list of specific guidelines for data entry, they are included in Appendix C to this report. Certain important design principles, however, do deserve to be emphasized in this general discussion, if only because some of these principles are so basic that they are seldom explicitly expressed.

Here is an example: an operator should not have to enter the same data twice. Now that is something every designer already knows, even if he does not often think about it. A corollary is

this: an operator should not have to enter a data item already entered by another operator. That seems to be just common sense, although one could imagine occasional exceptions to the rule when cross validation of data inputs may be required.

How can we avoid duplicative data entry in practice? The key lies in designing the MMI (i.e., programming the computer) to maintain context. Thus when an operator identifies a particular squadron of interest, the computer should be able to access all previously entered data relevant to that squadron and not require the operator to enter such data again. If an operator enters one item of data about a particular squadron, it should be possible to enter a second item immediately thereafter without having to re-identify that squadron. In repetitive data entry transactions the operator should have some means of defining default entries for selected data items, in effect telling the computer those items will stay the same until the default value is changed or removed.

Context should also be preserved to help speed correction of input errors. One significant advantage of on-line data entry is the opportunity for immediate computer validation of operator inputs, timely feedback so that an operator can correct detected errors while that set of entries is still fresh in his mind and/or while documented source data are still at hand. Here the computer should preserve the context of the data entry transaction, remembering correct items so that the operator does not have to enter those again while changing incorrect items.

The preservation of context is, of course, important in all aspects of man-machine interaction, with implications for data output, sequence control and operator guidance. The importance of context will be emphasized again in the further discussion of those other general functions.

Another important design concept is that of flexibility. The idea that the MMI should adapt flexibly to operator needs is often expressed. The means of achieving such flexibility should be spelled out in MMI guidelines. For data entry functions it is important that the pacing of inputs be controlled flexibly by the operator. Tasks where the pacing of operator inputs is set by the machine (for example, ZIP code keying at "automated" post offices) are stressful and error-prone.

Aside from flexibility in pacing, the operator will generally benefit from having some choice in the ordering of inputs. Although this kind of flexibility is related to the topic of sequence control, it merits discussion here as well. What is needed in MMI

design is some sort of suspense file(s) to permit flexible ordering of operator inputs, including temporary omission of unknown items, backup to correct mistaken entries, cancellation of incomplete transactions, etc.

As suggested earlier, the data input operator may also benefit from flexibility in defining his own default options to simplify entry during a sequence of transactions. Some systems include only those defaults anticipated by the designers, which may not prove helpful to the operator in a particular instance.

It is obvious that design of data input transactions is necessarily dependent on hardware selection. For that reason, design guidelines for input devices receive considerable attention. A notable example is standardization of keyboard layouts. For further comment see Table A-18 in Appendix A.

Future technological advances in input hardware may well influence the design of data entry tasks, presaged perhaps by the current advocacy of voice input. But the major need in C3 systems is for consistently good software design. It is in improving the logic of data entry that the chief gains can be made, and it is here that design guidelines should prove most helpful.

DATA DISPLAY/OUTPUT

Data display, i.e., some kind of output from the machine to the operator, is needed for all C3 tasks. Data display is emphasized particularly in monitoring and control tasks. Included under this heading may be hardcopy printouts as well as more mutable electronic displays. Also included are any auxiliary displays or signaling devices, including voice output, used to alert the operator of unusual conditions. Displays specifically intended to guide the operator in his interaction with the system are discussed later under the topics of sequence control and operator guidance.

In general, it may be said that rather less is known about data display and information assimilation by the operator than about data entry. In present C3 system design, display formatting is an art. Guidelines are surely needed.

Here again some general concepts deserve emphasis, including the importance of context and flexibility. Data displays must always be interpreted in the context of the operator's task requirements and expectations. An early statement of the need for relevance in data display seems valid still:

> When we examine the process of man-computer communication from the human point of view, it is useful to make explicit a distinction which might be described as contrasting "information" with "data." Used in this sense, information can be regarded as the answer to a question, whereas data are the raw materials from which information is extracted. A man's questions may be vague, such as, "What's going on here?" or "What should I do now?" Or they may be much more specific. But if the data presented to him are not relevant to some explicit or implicit question, they will be meaningless. This is also true, in some degree, for the computer, which is programmed to accept and process certain kinds of data and not others. However, the limitation in the case of the man is somewhat different, being more in the nature of a generally low processing rate or attention span.
>
> What the computer can actually provide the man are displays of data. What information he is able to extract from those displays is indicated by his responses. How effectively the data are processed, organized, and arranged prior to presentation will determine how effectively he can and will extract the information he requires from his display. Too frequently these two terms data and information are confused, and the statement, "I need more information," is assumed to mean, "I want more symbols." The reason for the statement, usually, is that the required information is not being extracted from the data. Unless the confusion between data and information is removed, attempts to increase information in a display are directed at obtaining more data, and the trouble is exaggerated rather than relieved.
>
> (Smith, 1963, pp. 296-297)

This distinction between data and information is still not familiar to display designers, although the issue itself is raised repeatedly. Consider the following description of our current "information explosion", and notice how using the terms data and information interchangeably tends to confound an otherwise incisive (and lively) analysis:

> The sum total of human knowledge changed very slowly prior to the relatively recent beginnings of scientific thought. But it has been estimated that by 1800 it was doubling every 50 years; by 1950, doubling every 10 years; and by 1970, doubling every 5 years. . . . This is a much greater growth rate than an exponential increase. In many

fields, even one as old as medicine, more reports have been written in the last 20 years than in all prior human history. And now the use of the computer vastly multiplies the rate at which information can be generated. The weight of the drawings of a jet plane is greater than the weight of the plane. The computer files of current IBM customer orders contain more than 100 billion bits of information -- more than the information in a library of 50,000 books.

For man, this is a hostile environment. His mind could no more cope with this deluge of data, than his body could cope with outer space. He needs protection. The computer -- in part the cause of the problem -- is also the solution to the problem. The computer will insulate man from the raging torrents of information that are descending upon him.

The information of the computerized society will be gathered, indexed, and stored in vast data banks by the computers. When man needs a small item of information he will request it from the computers. The machines, to satisfy his need, will sometimes carry on a simple dialogue with him until he obtains the data he wants. With the early computers, a manager would often have dumped on his desk an indigestible printout -- sometimes several hundred pages long. Now the manager is more likely to request information when he needs it, and receive data about a single item or situation on a screen or small printer.

It is as though man were surviving in the depths of this sea of information in a bathyscaphe. Life in the bathyscaphe is simple, protected as it is from the pressure of the vast quantities of data. Every now and then man peers through the windows of the bathyscaphe to obtain facts that are necessary for some purpose or other. The facts that he obtains at any one time are no more than his animal brain can handle. The information windows must be designed so that man, with his limited capabilities, can locate the data he wants and obtain simple answers to questions that may need complex processing.

(Martin, 1973, p.6)

Somehow we must find the means to provide and maintain context in data displays so that the operator can find the information he needs for his job. Task analysis may point the way here, indicating

what data are relevant to each stage of task performance. Design guidelines must emphasize the value of displaying no more data than the operator needs, maintaining consistent display formats so that the operator always knows where to look for different kinds of information, and using consistent labeling to help the operator relate different data items, on any one display and from one display to another.

Detailed operator information requirements will vary from time to time, however, and may not be completely predictable in advance, even from a careful task analysis. Here is where flexibility is needed, so that data displays can be tailored on-line to operator needs. Such flexibility is often provided in C3 systems through category selection, optional display offset and expansion features. If such options for display coverage are available, the operator may be able to adjust his information processing of data outputs in a way analogous to self pacing of data inputs.

In tasks where an operator must both enter and retrieve data, which is often the case, the formatting of data displays should be compatible with the methods used for data entry. As an example, if data entry is accomplished via a form-filling dialogue with a particular format for data fields, subsequent retrieval of that data set should produce an output display with the same format, especially if the operator is expected to make changes (and/or additional entries) to displayed data. Where compaction of data output is required for greater efficiency, to review multiple data sets in a single display frame, the displayed items should retain at least an ordering and labeling compatible with those fields used previously for data entry.

Display design should also be compatible with dialogue type(s) and hardware capabilities, as noted earlier. Where operator inputs are made via menu selection, using a pointing device like a lightpen, then display formats should give prominence (and adequate separation) to the labeled, lightpennable options. Location of multi-function keys at the display margin, to be labeled on the adjacent portion of the display itself, may provide flexibility for both data entry and sequence control, but will necessarily constrain the formatting of displays for data output.

For further comment on data display see Tables A-14 through A-17 in Appendix A.

SEQUENCE CONTROL

Sequence control refers to the logic and means by which inputs and outputs are linked to become coherent transactions, and which

govern the transitions from one transaction to the next. Techniques of sequence control require explicit attention in MMI design, and a number of published guidelines bear on this topic. The general ideas of flexibility and context are important in sequence control as in other aspects of MMI design.

Ideal flexibility would permit the operator to undertake whatever task or transaction he wishes, at any time. Although this may not always prove feasible, the MMI designer should try to provide the maximum possible user control of the on-line transaction sequence. As a simple example, suppose the operator is scanning a multi-page data display. He should be able to go either forward or back at will. If the MMI design only permits him to step forward, so that he must cycle through the entire display to reach a previous page, that design is deficient. The operator should also be able to interrupt display scanning at any point to initiate some other transaction. Such simple flexibility is relatively easy for the designer to achieve, and indeed is commonly provided.

More difficult are transactions that involve potential change to stored data. Here again the operator will need some flexibility in sequence control, perhaps wishing to back up in a data entry sequence to change previous items, or to cancel and restart the sequence, or to abort the sequence altogether and escape to some other task. The MMI designer can provide such flexibility through use of suspense files, as suggested in the earlier discussion of data entry, and other special programmed features. This flexibility requires extra effort from the designer and programmer. But that extra effort is made only once, and is a worthwhile investment on behalf of the eventual operators who may interact with their computer system for months or years.

In one sense, flexibility of sequence control has pitfalls. Just as an operator may make a mistake in data entry, so also can he make a mistake in sequence control. The MMI designer must try to anticipate operator errors in sequence control and ensure that potentially irreversible actions are difficult to take. In data entry tasks, for example, when an operator is satisfied with a set of keyed data he should be obliged to take some explicit action to ENTER it for computer processing. The MMI should be designed to protect the operator from the consequences of inadvertently destructive actions. Any large-scale erasure or deletion of data, for example, should require some sort of explicit operator confirmation, being accomplished as a two-step process rather than a single transaction. This provides a software analogy to the physical barriers sometimes used to protect critical hardware controls from accidental activation.

One form of flexibility frequently recommended is the provision of alternate modes of sequence control for experienced and inexperienced operators. In a command-language type of dialogue, optional guidance might be provided to prompt a beginner step by step in his composition of commands for sequence control, whereas an experienced operator might enter a complete command as a single complex input. Some such flexibility is surely desirable -- to interpret halting, stepwise commands as well as fluent, coherent inputs.

More generally, however, it may be desirable to include redundant modes of sequence control in MMI design, perhaps involving combinations of different dialogue types. As an example, menu selection might be incorporated to provide easy sequence control for beginners, but every display frame might also be formatted to include a standard entry field where an experienced operator could key in complete sequence control commands more efficiently. Examples of this approach have been provided by Palme (1979).

Another way to provide flexibility in sequence control is through specific tailoring of display formats. Consider, for example, a dialogue type in which sequence control is exercised through lightpen selection among displayed command options. For any particular display frame it might be possible to display those three or four options most likely to be selected by the operator at that point in the task sequence, with perhaps a general purpose OPTIONS selection that could be used to call out a display of any other (less likely) sequence control commands. Thus, on the first page of a two-page display, one of the likely sequence control commands would be NEXT PAGE; but on the second page that command would be replaced by its now more likely complement, PREV PAGE.

This approach illustrates two design ideas. The first is as close as we will probably come to a general rule for sequence control: make the operator's most frequent transactions the easiest to accomplish. The second idea is the reliance on context to improve flexibility in MMI design.

The importance of context in sequence control extends beyond the example above, where likely control options are given preferential display. Context can be used to shorten command inputs during sequence control and still permit unambiguous interpretation. In general, it will prove desirable to have the operator rather than the computer define the task context in which control inputs are to be interpreted.

As an example, suppose that an operator wishes to assign several sorties of aircraft from a particular squadron to preplanned missions. He should be able to specify the squadron and then expect all subsequent commands to be applied to that squadron. Further, he should be able to specify that he is making assignments and expect that subsequent selections of aircraft/crews and missions will be interpreted accordingly. If context can be used in this way, the operator will not have to take repetitious control actions, which is just as desirable as avoiding repetitious data entries.

A potential pitfall here is that the results of a particular control action, if contingent on context, may vary from one time to the next. Thus aircraft selection in the context described above would result in mission assignment, whereas aircraft selection in a different context might result in unassignment, or allocation to ready reserve, or perhaps commitment to maintenance status. Such variability may prove confusing to an operator. If the consequences of control actions are made contingent on context, then the MMI designer should ensure that the current context and its implied contingencies are always displayed to the operator. That will help maintain operator orientation as discussed next.

USER GUIDANCE

Many MMI design features contribute directly or indirectly to guide the user in his interaction with an on-line computer system. A general discussion here may serve to illustrate the range of factors that should be considered.

The primary principle governing this aspect of MMI design is to maintain consistency. Design consistency is emphasized in all published guidelines. With consistent MMI design, the operator will learn to use his computer tool more quickly and with more confidence. Consistency in the small details of MMI operation will help serve as an antidote, or stable counterbalance, to the functional flexibility advocated above. Without such consistency of detail, flexibility in MMI design may make a system easy to use by an expert but hard to learn for a beginner.

Design consistency implies predictability of system response to operator inputs. A fundamental rule is that some response be received: for every action (input) by the operator there should be some noticeable reaction (output) from the machine processor. It is this feedback linking action to reaction that defines each discrete transaction and maintains operator orientation in his interaction with the system. In the absence of response, the operator will become frustrated and uncertain of proper machine functioning.

A rule such as this is so general in its application to MMI design that it is seldom stated explicitly. Because it would apply to any on-line information handling task, it might not be called out in a task-specific requirements matrix. There may be some other design guidelines that also have this kind of general applicability. Such general guidelines could be referenced in the functional MMI specification for any C3 system acquisition, in addition to the more specific design guidelines derived from task analysis.

Predictability of machine response also relates to system response time. Timely response can be critical in maintaining operator orientation to his task. If a response is received only after a long delay, the operator's attention may have wandered. Indeed, he may forget just which of his actions the machine is responding to. Frequent operator actions, generally those involving simple inputs such as sequence control selections, should be acknowledged immediately. In transactions where output must be deferred pending the results of computer search and/or calculation, the expected delay should be indicated to the operator in a quick interim message.

Some experts have argued that consistency of system response time may be more important in preserving operator orientation than the absolute value of the delay, even suggesting that designers should delay fast responses deliberately in order to make them more consistent with necessarily slow responses. To some extent such a reduction in response time variability may be desirable. If system response time is always slow, an operator may adapt to the situation and find something else useful to do while waiting. But surely the better solution is to make all responses uniformly fast, or where that is not possible to signal quickly the occasional slow response as suggested above. In this way, an unusually slow response can be made predictable although inconsistent.

As it happens, variability in response time is probably a greater problem in the design of general-purpose, multiple-user, time-shared commercial systems than for most C3 applications. Where C3 systems are designed to accomplish defined tasks in a predictable manner, data processing loads can usually be anticipated sufficiently well in MMI design to provide adequately quick response time.

Consistency is also important in other aspects of MMI design. Display formats should be consistent from one frame to another, including always a title at the top to tell the operator where he is, labels to indicate page number in related displays, standard labeling of control options, standard positioning of guidance

messages, etc. Messages indicating operator error should be carefully worded to be both concise and informative, consistently worded from one message to the next, and consistently located in the display format.

Even such a subtle feature as cursor positioning at display generation should receive consistent treatment in MMI design. Consistent positioning is particularly helpful to the operator if the cursor does not have a prominent appearance and/or the means of manual cursor positioning are inconvenient. At display generation the cursor should be placed in its most probably useful position, such as a data entry field. If input errors are detected, the cursor should be placed in the first data entry field requiring correction in the regenerated error display.

Data entry is also facilitated by consistent treatment of data entry fields on the display, including consistent wording of labels, consistent placement of labels with respect to entry fields, and consistent demarcation of the fields themselves. Underlining is frequently recommended for field delineation, displaying clearly to the operator where each field is located and also its defined extent. Even more guidance could be provided by consistent use of different underlines to indicate different types of data entry, perhaps dots where optional entries can be made, and dashes where required entries are expected.

For sequence control, consistent guidance will also prove helpful to the operator. As a general principle, the MMI designer should not rely on the operator to remember what he has just done, or even what he is currently doing. Operators are often distracted by competing job demands. For extended transaction sequences the designer should arrange to display a cumulative record of control actions for operator review. For control actions whose consequences are contingent on context, an indication of that context should always be displayed even if context was defined by the operator, as recommended in the earlier discussion of sequence control.

Another design feature that can help guide the operator is consistent provision of an OPTIONS display, a "home base" to which the operator can return from any point in the control sequence in order to select a different transaction.

Although consistent MMI design will provide much inherent guidance to the operator, it is often also desirable to include in computer-generated display outputs some explicit instructions or reminders to the operator. Such instructions should be consistently located in display formatting, perhaps always at the bottom where

they can be ignored by experienced operators. For inexperienced operators it may be desirable to provide supplementary guidance or job instruction, optionally available in response to operator request, perhaps through selection of a HELP or EXPLAIN option. Such optional guidance will help adapt the MMI to different operator capabilities, supporting the novice user without hindering the expert.

It should be clear that a general discussion of the kind provided here offers some perspective to the MMI designer, but little direct guidance. General principles must be elaborated in specific guidelines, as illustrated by the sample set for data entry functions proposed in Appendix C. Further, there must be some means of comparing proposed designs with those guidelines, as discussed in the next section of this report.

SECTION 4

DESIGN REVIEW AND VERIFICATION

Having decided what is needed in MMI requirements definition, and having specified what is needed in design guidelines, the third step is to ensure that what is specified will actually be included in the MMI design. Some review process should be established to permit early verification of a proposed MMI design before its software implementation, and continuing design coordination thereafter.

DESIGN DOCUMENTATION

The key to effective design review lies in documentation. In system acquisition it has become customary to require documentation in advance of design, at least for critical design elements. In systems where MMI design is considered critical, which probably includes most C3 systems, documentation should describe interface characteristics influencing software design as well as hardware selection, including for each on-line transaction the expected inputs, required data processing, format of displayed outputs, and the associated sequence control logic.

There are several ways in which documentation of MMI design could be specified. First, a standard Data Item Description (DID) might be interpreted to require system development contractors to provide MMI design data. The most obvious candidates would be the current (1979) DIDs for human engineering design, DI-H-7056 for system operation and DI-H-7057 for maintenance. Although these DIDs are primarily oriented toward equipment design and workspace layout, there is some implied acknowledgment of software design: "Display symbology, display formats and control/display operation logic shall be described with regard to intended use by the operator(s)" (DID-H-7056, 1979, page 2). Further, there is a stated requirement to include "narrative which provides rationale for any need to deviate from, or take exception to, MIL-STD-1472 or other contractual human engineering documents" (DI-H-7056, 1979, page 3). Presumably any contractually-imposed MMI design guidelines might be considered as falling in this "other" category.

A second approach might be to modify an existing DID to deal more specifically with MMI design. It is evident from the meager quotations cited above that DI-H-7056 provides little specific guidance to a contractor as to just what should be included in the

description of MMI design. But that DID could be expanded for a particular system acquisition to indicate more clearly what is needed.

A third approach might be to draft a new contractual requirement dealing specifically with MMI design, thus creating a so-called Unique Data Item (UDI). This approach might serve for initial exploration of the usefulness of MMI design guidelines, but in the long run a more standardized approach will be desirable so that a new UDI does not have to be created for each new program. Standardized documentation requirements in the long run should accommodate the needs of all military services, since there is a strong emphasis within DoD toward eventual adoption of unified procurement standards.

Whatever its eventual designation, the contractual requirement for contractor documentation of MMI design should probably specify both the content and the format for description of the MMI. With little precedent for such documentation, one example is the approach proposed by Pew and Rollins (1975) to document display formats, data inputs, operator actions and their consequences. This approach to MMI design documentation is discussed further in Appendix D.

DESIGN REVIEW

The purpose of MMI design documentation is review and coordination. Design review offers several potential advantages. Early review would allow critique of inadequate design features before actual implementation. At this stage, suggestions for improvement of MMI design would not entail the costs and delays of software modification. To permit effective review, the contractor should document the proposed MMI design so that it is clearly explained to both system developers and intended users.

In some system acquisitions the contractor may be encouraged to propose additional MMI design standards beyond those included in the system specification. Standards proposed by the contractor should be included in MMI design documentation, so that design review can determine whether MMI functional requirements will be met.

In an acquisition program involving modifications to improve an existing system, rather than development of a completely new system, the contractor may be asked to define MMI guidelines consonant with current procedures, which represent *de facto* design standards. Again, MMI design documentation will demonstrate contractor understanding of functional requirements. With increased emphasis on incremental procurement in system acquisition, modification and

expansion of existing system capabilities may become the typical mode of future ESD system development programs.

One aspect of MMI design review deserves particular attention. In many C3 systems the MMI represents such a significant and pervasive element of the total effort that review of MMI design can be a large first step in reviewing overall system design. If operators are to be intimately involved in critical system functions, then a detailed review of proposed operator functions will go a long way toward understanding system functions. Thus a document describing MMI design could, with some amplification to include internal data processing and interfaces plus relevant external constraints, become a description of the larger information system in which the operators will work.

Such reference to MMI design in order to clarify understanding of more general system operation can prove helpful in system acquisition. In ESD programs there is evidence that MMI design documents have sometimes been generated informally, outside of contractual requirements, as a spontaneous aid to overall design review and coordination in system development.

DESIGN COORDINATION

Documentation of MMI design could serve a variety of purposes in design coordination. During initial software implementation, an MMI "design handbook" could provide a single reference to help coordinate the contributions of different people in hardware and software engineering groups. Such a handbook could also provide the basis for configuration management of MMI design, as subsequent changes are proposed to initial implementation.

For the purpose of coordination, MMI design documentation should include a summary of the guidelines actually used, i.e., as adapted from initial design proposals to take into account subsequently imposed software tradeoffs, hardware constraints, etc. In effect, the MMI designer should document his work as if he were telling someone else how to do it.

Such explicit documentation of system-specific MMI guidelines is presently rare, but is nonetheless needed. A recent example is provided in a NASA report of MMI guidelines providing a sort of design handbook for Spacelab experiments:

> The purpose of this document is to present CRT display design and command usage guidelines applicable to experiments utilizing the Experiment Computer Application

> Software (ECAS) for use by Spacelab scientific investigators and experiment designers. It is expected that most Spacelab experiments will have need of such displays.
>
> The guidelines, while not given as strict requirements, explain recommended methods and techniques for presenting data from the ECAS program via the DDS data display system to the payload crew. The user is encouraged to apply and follow them, although other considerations (memory conservation, etc.) may force a trade-off in specific instances. Used as a reference, the document will be an important aid in standardizing the crew/experiment interface among the different payloads and should result in lower crew training time and increased efficiency of the payload crew in onboard experiment operations.
>
> (NASA, 1979, page 1-1)

An MMI design handbook could also provide timely information to coordinate the preparation of operator manuals and training materials in parallel with design implementation, in the course of system development.

MMI design documentation could also continue to provide helpful guidance after system implementation, for users who may wish to add or modify displays, operator inputs, sequence control logic, etc., in order to meet changing operational requirements. That may become more common in the future as existing C3 systems are upgraded rather than new systems built. In such situations, it could be important to maintain consistency in MMI design between "old" and "new" portions of a system.

It may seem redundant to keep a written desciption of MMI design once a system is operating. Why not just look at the actual on-line interface to see what it is like? Certainly it is true that operational use can provide valuable insight into the advantages and drawbacks of any particular MMI design. And it is operational performance that provides final design verification. But operational use is a slow and inefficient way to review some design features, for example, to discover system response to inputs seldom made and transaction sequence paths seldom taken.

Moreover, the principles underlying MMI design, the functional requirements and the chosen guidelines for meeting them, will probably not all be evident in operational use. It may well be easier for future designers and users if those principles can be set

down explicitly for each system, as well as more generally for all systems. Which brings this report full circle in its recommendations: system analysts should define and document functional MMI requirements as guidelines for the designer, and MMI designers should in turn document for the system user what they have done.

SECTION 5

CONCLUSIONS AND RECOMMENDATIONS

This report proposes that a significant program of work be undertaken. It marks a beginning, not an end. Thus only tentative, interim conclusions are offered, and the chief recommendation is that the work be carried forward. To follow the approach outlined here, the next steps involve further development of tools for MMI requirements definition and design, and evaluation of the application of those tools in C3 system acquisition.

Three kinds of tools have been proposed:

1. An MMI requirements matrix, a systematic tabulation of the functional capabilities required by different operator tasks, to permit requirements definition in advance of MMI design. The concept of a requirements matrix is described in Section 2 of this report and illustrated in Appendix B. A tentative conclusion is that this requirements matrix could prove a useful tool, at the least providing a check list for systematic consideration of MMI requirements, with some potential promise as an effective means of coordinating analysis with design. Further development of this requirements matrix is recommended in conjunction with evaluation in system acquisition.

2. MMI design guidelines, a compilation of available wisdom, to be tailored to the requirements of different system applications. Design guidelines are discussed in Section 3 of this report and illustrated in Appendix C. A tentative conclusion is that guidelines can be found for the design of certain common MMI functional capabilities, and that these can be related at least approximately to the requirements matrix, with potential for tailoring. It is recommended that the sample set of guidelines presented here be expanded to provide broader functional coverage, and then evaluated in system acquisition.

3. Design documentation, some means of specifying detailed MMI design for coordination and review in advance of software implementation. Design

documentation is discussed in Section 4 of this report, and one possible approach is illustrated in Appendix D. A tentative conclusion is that such documentation could prove useful for both initial design and subsequent design modification, but that it is not clear just what are the proper means for imposition of this special documentation requirement in system acquisition. This question should be explored further, until a formal documentation requirement can be developed.

All of these proposed tools must be applied to assess their value. Discussion of the possible utility of these tools for MMI design is an interesting exercise in armchair philosophy, but will have no practical effect unless carried forward and tested in actual system development. Certainly no Air Force or DoD design standard can be justified on the basis of this discussion alone, without practical evaluation.

In Section 1 of this report, a long introduction of background material concluded with a call for a collaborative effort, asking people responsible for system acquisition to join in the development and evaluation of tools for MMI design. A call for joint participation is appropriate again here at the conclusion of this report.

Are you involved in a system acquisition program where MMI design will be critical to effective system operation? Could an MMI requirements matrix help define needed functional capabilities? Would guidelines help MMI design and software implementation? Will formal documentation of MMI design help system development, evaluation and future operation? If so, consider using the tools proposed here.

REFERENCES

ARI (Army Research Institute for Behavioral and Social Sciences). Development of Design Guidelines and Criteria for User/Operator Transactions with Battlefield Automated Systems, RFP No. MDA903-79-R-0218, 1979.

Calhoun, G. L. Control logic design criteria for multifunction switching devices. In Proceedings of the 22nd Annual Meeting. Santa Monica, California: Human Factors Society, 1978, 383-387.

Chamberlain, R. G. Conventions for interactive computer programs. Interfaces, November 1975, 6(1), 77-82.

Chariton, D. R. Man-machine interface design for timesharing systems. In Proceedings of the Annual Conference. New York: Association for Computing Machinery, 1976, 362-366.

Clapp, J. A. and Hazle, M. Building Blocks for C3 Systems, ESD-TR-77-360, Electronic Systems Division, AFSC, Hanscom AFB, MA, March 1978, ADA052568.

DI-H-7056. Data Item Description: Human Engineering Design Approach Document - Operator. Washington: U.S. Government Printing Office, 1 June 1979.

DI-H-7057. Data Item Description: Human Engineering Design Approach Document - Maintainer. Washington: U.S. Government Printing Office, 1 June 1979.

Dzida, W., Herda, S. and Itzfeldt, W. D. User-perceived quality of interactive systems. In Proceedings of the 3rd International Conference on Software Engineering, IEEE Catalog No. 78CH1317-7C, 1978, 188-195.

Engel, S. E. and Granda, R. E. Guidelines for Man/Display Interfaces, Technical Report TR 00.2720. Poughkeepsie, New York: IBM, December 1975.

Foley, J. D. and Wallace, V. L. The art of natural graphic man-machine conversation. In Proceedings of the IEEE, April 1974, 62(4), 462-471.

Goodwin, N. C. Man-machine interface characteristics. Published as an Appendix in J. A. Clapp and M. Hazle, Building Blocks for C3 Systems, ESD-TR-77-360, Electronic Systems Division, AFSC, Hanscom AFB, MA, March 1978, ADA052568.

Green, E. E. Message design -- graphic display strategies for instruction. In Proceedings of the Annual Conference, New York: Association for Computing Machinery, 1976, 144-148.

Hansen, W. J. User engineering principles for interactive systems. In AFIPS Conference Proceedings, 1971, 39, 523-532.

Kennedy, T. C. S. The design of interactive procedures for man-machine communication. International Journal of Man-Machine Studies, 1974, 6, 309-334.

Krebs, M. J. Design principles for the use of color in displays. In SID International Symposium Digest of Papers. Los Angeles: Society for Information Display, 1978, 28-29.

Martin, J. Design of Man-Computer Dialogues. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

MIL-H-48655B. Military Specification: Human Engineering Requirements for Military Systems, Equipment and Facilities. Washington: Department of Defense, 31 January 1979.

MIL-STD-454F. Military Standard: Standard General Requirements for Electronic Equipment. Washington: Department of Defense, 15 March 1978.

MIL-STD-483. Military Standard: Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs. Washington: Department of Defense, 31 December 1970.

MIL-STD-1472B. Military Standard: Human Engineering Design Criteria for Military Systems, Equipment and Facilities. Washington: Department of Defense, 31 December 1974.

Miller, R. B. Response time in man-computer conversational transactions. In AFIPS Conference Proceedings, 1968, 33, 267-277.

NASA (National Aeronautics and Space Administration). Spacelab Experiment Computer Application Software (ECAS) Display Design and Command Usage Guidelines, Report MSFC-PROC-711. George C. Marshall Space Flight Center, Alabama, January 1979.

Nickerson, R. S. and Pew, R. W. Oblique steps toward the human-factors engineering of interactive computer systems. Published as an Appendix in M. C. Grignetti, D. C. Miller, R. S. Nickerson and R. W. Pew, Information Processing Models and Computer Aids for Human Performance, Report No. 2190. Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc., June 1971. (NTIS No. AD732913)

Palme, J. Interactive Software for Humans, Report FOA-C10029-M3(E5). Stockholm: Swedish National Defense Research Institute, July 1975.

Palme, J. A human-computer interface for non-computer specialists. Software - Practice and Experience, 1979, 9, 741-747.

Penniman, W. D. Past chairman's message. SIG Newsletter No. UOI-10. Washington: American Society for Information Science, May 1979.

Pew, R. W. and Rollins, A. M. Dialog Specification Procedures, Report 3129 (revised). Cambridge, Massachusetts: Bolt Beranek and Newman, 1975.

Pew, R. W., Rollins, A. M. and Williams, G. A. Generic man-computer dialogue specification: an alternative to dialogue specialists. In Proceedings - 6th Congress of the International Ergonomics Association. Santa Monica, California: Human Factors Society, 1976, 251-254.

Ramsey, H. R. and Atwood, M. E. Human Factors in Computer Systems: A Review of the Literature, Technical Report SAI-79-111-DEN. Englewood, Colorado: Science Applications, Inc., September 1979. (NTIS No. ADA075679)

Ramsey, H. R., Atwood, M. E. and Kirshbaum, P. J. A Critically Annotated Bibliography of the Literature of Human Factors in Computer Systems, Technical Report SAI-78-070-DEN. Englewood, Colorado: Science Applications, Inc., May 1978. (NTIS No. ADA058081)

Rouse, W. B. Design of man-computer interfaces for on-line interactive systems. In Proceedings of the IEEE, June 1975, 63(6), 847-857.

Smith, S. L. Angular estimation. Journal of Applied Psychology, 1962, 46, 240-246.

Smith, S. L. Man-computer information transfer. In Howard, J. H. (Ed.) Electronic Information Display Systems, 284-299. Washington: Spartan Books, 1963.

Smith, S. L. An On-Line Model of Traffic Control in a Communications Network, Technical Report MTR-2813. Bedford, Massachusetts: The MITRE Corporation, March 1974.

Smith, S. L. MAC Air Cargo Data Entry: 1. Preliminary Analysis of Alternative Modes, ESD-TR-76-162, Vol. I, Electronic Systems Division, AFSC, Hanscom AFB, MA, August 1976a, ADA029013.

Smith, S. L. MAC Air Cargo Data Entry: 5. Field Testing an On-Line Terminal at the Dover Aerial Port, Technical Report MTR-3066-5. Bedford, Massachusetts: The MITRE Corporation, 30 June 1976b.

Smith, S. L. and Goodwin, N. C. Computer-generated speech and man-computer interaction. Human Factors, 1970, 12(2), 215-223.

Smith, S. L. and Goodwin, N. C. Alphabetic data entry via the Touch-Tone pad: A comment. Human Factors, 1971, 13(2), 189-190.

Stewart, T. F. M. Displays and the software interface. Applied Ergonomics, 1976, 7(3), 137-146.

Thompson, D. A. Interface design for an interactive information retrieval system: a literature survey and a research system description. Journal of the American Society for Information Science, 1971, 22, 361-373.

Uber, G. T., Williams, P. E. and Hisey, B. L. The organization and formatting of hierarchical displays for the on-line input of data. In AFIPS Conference Proceedings, 1968, 33, 219-226.

Wasserman, A. I. The design of 'idiot-proof' interactive programs. In AFIPS Conference Proceedings, 1973, 42, M34-M38.

Williams, T. J. (Ed.) Guidelines for the Design of Man/Machine Interfaces for Process Control. West Lafayette, Indiana: Purdue University, January 1977. (NTIS No. ADA036457)

APPENDIX A

SUMMARY INFORMATION RELATING TO MMI DESIGN

This appendix presents tables taken from the report on human factors in computer systems by Ramsey and Atwood (1979). They are reprinted here by permission. These tables summarize information relating to various aspects of MMI design, and thus provide a useful supplement to various topical discussions in the text of this report.

The tables have been cited in the text, but are presented here separately to emphasize their independent provenance. These tables preserve the titles provided by their authors, and are in the same general order, but their original "Figure" numbers are changed somewhat here. The Ramsey and Atwood report, of course, contains much more detailed information than can be included in these tables, and should prove an important source document for MMI designers.

An explanation is needed of the reference notation used in the tables. The listed references are cited more completely at the end of their source report (Ramsey and Atwood, 1979), and can also be found in a separately published annotated bibliography (Ramsey, Atwood and Kirshbaum, 1978). References marked with a single asterisk indicate reports of survey, questionnaire, or summarized data. A double asterisk indicates a report of performance data or detailed results of experimental studies.

Table A-1

Properties of Some Major Classes of Interactive System User

User Class	Discussion	Principal References
Naive users (inexperienced with computers)	Computer-naive users are actually a very heterogeneous group, but have many common properties. Naive users benefit greatly from computer-initiated dialogue, usually require more tutorial features. Correct implicit "mental model" of computer systems and interactive dialogue cannot be assumed, must be explicitly conveyed by system. Naive user population has many detailed implications for dialogue design. Smooth transition from naive to experienced user is often difficult in current systems.	Card et al (1974)** Eason et al (1975)* Evans (1976) Martin (1973) Nickerson & Pew (1971) Thompson (1969, 1971)
Managers (including military commanders, etc.)	Managers tend to have highly variable information needs; current systems are often too rigidly constraining to satisfy those needs. Managers tend to place high negative value on own effort, have considerable discretion with respect to mode of system use or nonuse. Thus, very low "impedance" is required to capture manager as direct user. If dissatisfied, manager tends to resort to "distant use" (interposing operator between manager and system) or partial use.	Ackoff (1967) Alter (1977)* Eason (1976)* Igersheim (1976) Raichelson (1972)
Scientific and Technical	High proportion report dissatisfaction with available automated tools. These users often respond to such dissatisfaction by becoming personally involved in design or implementation of software tools, or by altering task to match available tools.	Stewart (1974, 1976b)*

* Presents survey or questionnaire data or summarizes experimental results.
** Presents user performance data or results of controlled experimental study.

Table A-2

User Responses to Inadequate System

Response	Definition	Comments
Dis-use	Reliance on other information sources.	Requires existence of alternative information source, user with sufficient discretion.
Mis-use	"Bending the rules" to shortcut operational difficulties.	Requires significant knowledge of system. May negatively impact system integrity.
Partial use	Use of (perhaps inappropriate) subset of system capabilities.	Users frequently adopt "satisficing" strategy, may not learn most relevant system capabilities.
Distant use	Interposition of operator between user and system.	Requires high status and discretion. Typical response of managers.
Modification of task	Changing the task to match capabilities of system.	Prevalent when tools are rigid, problem is unstructured, as in scientific problem solving.
Compensatory user activity	Compensation for system inadequacies by additional user actions.	Typical with users of low discretion, as clerks.
Direct programming	Programming by user, in order to modify system capabilities to suit needs.	Typical response of computer-sophisticated user, as scientific and engineering user.
Frustration and apathy	Response of user when above actions are inadequate or unsatisfactory.	Involves lack of user acceptance, high error rates, poor performance.

Table A-3

Representative Examples of Statistical Studies of User Behavior in Time-Sharing Systems

System	Data Reported	Reference
JOSS	Statistics on input/output quantities and rates, user session temporal properties, space/time properties of executed programs.	Bryan (1967)**
TSS/360 Program Control System at IBM research center	Statistics on frequencies of use of commands by class (e.g., data management, editing, debugging) and specific command. Discussion of user competence groups.	Boles & Spiegel (1973)**
IBM/360 Time Sharing System	Statistics on duration and frequency of terminal sessions, use of language processors, command usage, and user response time.	Boles (1974)**
IBM/360 Batch Processing System	Statistics on utilization of compilers and system resources, temporal distribution of jobs and CPU utilization by day, week, and year	Haralambopoulos & Nagy (1977)**
AUTOGRP medical statistical analysis system	Frequencies of use of commands overall and by user. Comparison of command usage for beginners and experienced users.	Carlisle (1972)**
TENEX time-sharing system	Statistics on session duration, CPU utilization	Grignetti et al (1971)**
MULTICS system	Frequencies of use of system commands, with statistics on associated "think times." Distributions of think time.	Rodriguez (1977)**

Table A-4

Requirements Analysis: Questionnaire and Survey Methods

Approach	Comments		References
Use of questionnaires to obtain ratings of the relative importance of various categories of information and system features.	Inexpensive. Difficult to be specific enough for detailed design decisions. Requires prior knowledge of all relevant information categories, although Delphi techniques might avoid this requirement.	An overriding limitation of questionnaire and survey techniques is their dependence on the correctness of the user's perception and description of the job, information requirements, work habits, etc. The user is expert at doing a job, not describing it. These subjective ratings often fail to correspond to objective performance comparisons, and may contain significant, if unintentional, biases and oversights.	Martin et al (1973)* McKendry et al (1973)
Use of questionnaires to obtain estimates of time spent on each task associated with recipient's job.	Self-estimates of time spent on work activities are notoriously poor. If only relative time is required, this may be adequate.		Dever (1972)*
"Repertory Grid Technique", a questionnaire-based technique for determining user's "cognitive frame of reference".	Difficult to use successfully. High-level, and may not easily be made specific enough for detailed design decisions. Might be more useful for "personalized" systems than for capturing requirements of broad user class.		Peace & Easterby (1973)*
"Delphi Technique", a survey technique in which recipient's responses are fed back, anonymously. Recipient responds again, while aware of previous responses of entire group.	Relatively expensive. Promotes consensus and identification of all information categories, but may suppress important individual differences. No instances found of application in specific area of computer systems design.		Linstone & Turoff (1975)
"Policy Capture", one of several techniques for developing quantitative relationships between perceived system desirability and specific system features. In this case, relationship takes the form of a multiple-regression equation.	Somewhat expensive. Mathematical assumptions may be inappropriate. Paired-comparison procedure limits dimensionality. No instances found of application in specific area of computer systems design.		Stewart & Carter (1973) (for discussion of other such techniques, see Cochrane & Zeleny, 1973)

Table A-5

Requirements Analysis: Interviews and Field Observation

Approach	Comments	References
Interviews with users to determine information requirements, decision points, organizational constraints, etc.	Used more or less universally. Formal discussion in literature is mostly in the context of management information systems. Has many variants, such as "structured" interviews. Although a skilled interviewer can overcome some of the limitations of subjectivity and inability of users to verbalize their practices, these limitations are still significant. To apply this method at the detailed system design level requires an insightful user, or interviewer, or both. Most useful for preliminary data collection.	Bentley (1976) King & Cleland (1975) Parker (1970)
"Ad Hoc Working Group", in which subject-matter experts devise system requirements by analysis and negotiation.	Appears somewhat effective at very high (undetailed) level. Has problems of subjectivity, and is susceptible to bias due to interpersonal relationships of group members (e.g., undue influence of high-status member). Probably irrelevant at detailed system design level.	Vandersluis (1970)*
"Critical Incident Technique", in which users are asked, via interview or survey, for information about incidents of particular success or failure in the process of which the computer system will be a part.	A broadly useful technique which often yields significant insights into critical functions and information. Often used by human factors personnel, but not evident in the computer systems design literature.	Flanagan (1954)
Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation.	Readily applicable to manual and clerical tasks, in which direct observation yields necessary raw data. Much more difficult to apply to cognitive tasks. Nonetheless, these techniques are broadly useful. "Task analysis" is often employed informally with inadequate detail and without necessary training in the technique. See also section on "Tasks and Task Properties".	Gallitz & Laska (1969, 1970)* Wylie et al (1975)* (for general discussion, see also R. M. Miller, 1962)

Table A-6

Requirements Analysis: Simulation and Gaming

Approach	Comments	References
"Paper" simulation, in which the possible function of a computer system is simulated by human observers, in order to obtain information about the user's problem-solving and information-seeking behavior.	Relatively inexpensive, and often very informative. Can be used in very unstructured form, freeing user from dialogue constraints which might interfere with problem solving, or in very structured form to simulate exact properties of proposed system. Can be obtrusive, and requirement for manual computation can cause unacceptably slow simulation.	Cross (1969)* Ramsey (1975) Van Cott & Kinkade (1968)
"Protocol analysis", in which the user comments extensively on his activities during simulated problem solving, and formal content analysis of the resulting commentary ("protocol") is used to make inferences about user behavior and problem-solving processes.	Similar in many respects to paper simulation, but more obtrusive, more detailed. Transcription and scoring of protocols is very time-consuming, often restricts use to a small number of cases. This is the indicated approach when a detailed problem-solving model is the goal, but it has also been used, with reported success, in detailed dialogue design.	Carlisle (1973) Heafner (1975)** Mann (1975)
Interactive simulation or gaming, in which the actual system, or an interactive computer simulation of the system, is used with a contrived scenario to observe user behavior and system performance.	If done during initial design phase, this approach requires sophisticated software tools or expensive software development. It is very effective in identifying design defects and improving dialogue. Requires prior system design, and is, therefore, more appropriate for design evaluation and iteration than for initial inquiry into user requirements.	Lenorovitz & Ramsey (1977) Martin & Parker (1971) Proctor (1963) (for a broad discussion of gaming, see Shubik, 1975)

Table A-7

Major Approaches to the Modeling of Interactive Systems

Approach	Description	Comments	Principal References
Network Models	These models treat user and system as equivalent elements in the overall process. The individual tasks performed by both the user and the system are described in terms of expected performance and in terms of logical predecessor-successor relationships. The relationships define a network of tasks which is used as a performance model of the user-computer system. Such models are usually used to predict either the probability of failure or "success", or the completion time, of an aggregate set of tasks.	Network models allow performance data about user and computer system to be integrated in a single model even though original data came from a variety of sources. However, performance data must be provided for each task, as must rules for combining performance data from individual tasks to obtain aggregated performance predictions. This is often difficult because of questionable or lacking empirical data, and because performance interactions among tasks (especially cognitive tasks or tasks performed in parallel) may be very complex. Performance distributions are often assumed without data. In spite of these difficulties, the process of constructing such models is often a valuable source of understanding.	Baker (1970) Pew et al (1977) Siegel et al (1973)
Control-theory Models	These models are based on control theory, statistical estimation, and decision theory. The user is regarded as an element in a feedback control loop. Such models are usually used to predict overall performance of the user-computer system in continuous control and monitoring tasks.	Control-theoretic models are more quantitative than other performance models. They may address user-computer communication broadly, but they ordinarily do not deal with details of the interface, such as display design. Therefore, their utility as an aid to the interactive system designer may be limited. Not much work has yet been done in applying these modeling techniques to situations in which the main user activities are monitoring and decision making, with infrequent control actions.	Pew et al (1977)

Table A-7 (Concluded)

Major Approaches to the Modeling of Interactive Systems

Approach	Description	Comments	Principal References
Decision Theory Models	These models concern the decision-making behavior of the user. They require the specification of: (1) a set of possible states of the world, with their estimated probabilities, and (2) a set of possible decisions, or courses of action, which might be taken, together with their expected values and costs in the various possible states of the world. Considering the values and costs, together with the evidence of particular world states, a decision-theoretic model can select courses of action.	Decision-theoretic models can be used to suggest "optimal" decisions or to describe the observed decision-making behavior of users. In both modes, these models are frequently used in decision aids. If it is reasonable to describe user behavior in terms of such a model, these models can also be useful to the system designer, as by suggesting information required by the user. However, such models have thus far been developed only for relatively simple, two-alternative decision situations. Most real-world tasks are more complex, and gross distortions may result from any attempt to express them in this simple form.	Pew et al (1977) (for a discussion of the use of these models in automated decision aids, see later section on "Problem-Solving Aids")
Models of Human Information Processing	In general, these models involve a characterization of: (1) the task environment, including the problem and means of solution available, (2) the problem space employed by the subject to represent the problem and its evolving solution, and (3) the procedure developed to achieve a solution. The method used to develop such models involves intensive analysis of the problem to be solved and of protocols obtained from problem solvers during solution.	Ideally, such efforts might lead to an integrative model of human information processing useable in a variety of design applications. However, existing models are either too task-specific for this use or are insufficiently detailed. Furthermore, relationships between task requirements and human performance capabilities and limitations are inadequately understood for human information processing tasks. There are many good models applicable to very specific tasks. They happen to be relevant to particular user tasks in the system undergoing design, and if the designer is sophisticated enough to recognize their relevance, these models may be useable, but this is asking a lot of the designer.	Mann (1975) Pew et al (1977) Ziegler & Sheridan (1969)
Computer System Models	These models attempt to describe the behavior of the computer component of an interactive system, but do not attempt to model user performance in detail. Some of the models do characterize user behavior in terms of the statistical properties of user commands for a particular application. The models usually attempt to predict such performance factors as system response time, CPU and memory loads, and I/O requirements.	These models tend to be relatively crude, but can be useful in determining whether or not user requirements with respect to response time and other gross system performance measures can be satisfied by a proposed system. They are of little assistance in determining what the user requirements are.	Carbonell et al (1968) Foley (1971) Grignetti et al (1971) Shemer & Heying (1969)

Table A-8

Basic Subtasks or Phases Involved in Problem Solving

Subtask or Phase	Description	Comments	Principal References
Problem Recognition	The first stage in problem solving is to recognize that a problem exists. People are frequently slow to recognize, or at least react to, problems. This is especially true in situations in which a person must monitor the current state of the environment and detect or react to critical changes.	A primary need is for an aid that alerts the problem solver to "relevant" changes in the environment. The relevant variables for a given task can be difficult to define. Current status displays, historical displays, and aids for dealing with degraded data can be useful. If the relevant variables are identified, coding techniques can be very useful.	Booth et al (1968) Chesler & Turn (1967) Scanlan (1975)** Smith, R. L. et al (1972) Topmiller (1968) Wylie et al (1975)
Problem Definition	After a problem is recognized, the problem solver must determine how to formulate, or represent, the problem. In most cases, there are several alternative formulations for a given problem. The overall success of problem solving strongly depends on selecting an appropriate formulation.	Aids that provide a change in problem representation (e.g., graphical displays, isomorphic representation) can be extremely useful. Developing alternative representations requires a thorough understanding of the specific problem and the problem-solving processes that are most appropriate. Allowing the problem solver to decompose a problem into subtasks and recombine these subtasks in various ways can be useful in problems with relatively independent tasks. This type of aid is less difficult to develop than changes in problem representation, but it is also less general.	Balzer & Shirey (1968) Cushman (1972) Krolak et al (1971) Newsted & Wynne (1976)* Smith, H.T. (1974)* Stewart (1974)
Goal Definition	In some cases, the goal to be achieved is pre-defined. In other cases, (e.g., tactical planning) the problem solver must select an appropriate goal. A selected goal must be not only appropriate, but also attainable	The primary difficulty is that a selected goal may not be attainable. It may be useful to aid the problem solver in generating several alternative, logically consistent goal structures and to delay selecting a specific goal until later in the problem-solving process. Research on goal definition is lacking.	None
Strategy Selection	Strategy selection is concerned with determining the general approach that will be used in problem solving. In some cases, a certain strategy is dictated by the problem representation that is selected. In general, strategy selection is based on previous experience with a given class of related problems.	The majority of strategy-selection aids are concerned with specific problem domains. This is appropriate since strategy selection is strongly driven by experience in a given domain. In domains in which problems can be decomposed into fairly independent subproblems, aids that allow the user to select strategies for these subproblems independently before combining them into an overall strategy can be very useful. Additional research is needed on the nature of specific problem-solving tasks and the strategy selection heuristics used by expert problem solvers. This would enable the construction of techniques to aid the less experienced user in this phase of problem solving.	Bennett (1971) Caruso (1970)* Wilde (1969)*

Table A-8 (Concluded)

Basic Subtasks or Phases Involved in Problem Solving

Subtask or Phase	Description	Comments	Principal References
Alternative Generation	In well-defined tasks, the problem solver can usually generate all alternative actions that may be appropriate. If there is a large number of alternatives, however, the problem solver may not be able to retain all alternatives in memory for later evaluation. If the task is not well-defined, the problem solver may not be able to generate appropriate alternative actions.	Aids that store a large number of user-generated alternatives can easily be developed and can also be effective. The principal need is for aids to suggest alternatives that the user is unable to generate. Such aids have been developed for training applications and for cases in which the computer has been programmed to generate optimal solutions without explicit user interaction. For ill-defined task environments, aids that suggest hypotheses to be tested may aid in alternative generation. Although potentially very useful, such aids could be difficult to construct.	Brown et al (1974) Carlson & Hodgson (1977)* Gagliardi et al (1965)* Hormann (1967)
Alternative Evaluation	Problem solvers are generally very good at evaluating alternatives in a manner consistent with their perception of the problem and the goal to be achieved. If the alternatives have far-reaching consequences or if they must be evaluated with respect to a large number of factors, the problem solver's memory and processing limitations may be exceeded.	In extremely well-defined task environments, aids have been developed that allow the user to simulate the consequences of various alternatives. Although they are very useful in specific cases, such aids have limited generality. Aids that capture the user's evaluation heuristics and then filter information to be consistent with these heuristics and sometimes even present alternatives considered to be optimal are especially useful when a large number of evaluation heuristics must be applied. This type of aid is both effective and general, but it requires a great deal of effort to implement.	Balzer & Shirey (1968) Brown et al (1975) Davis et al (1975)** Doutriaux (1973)* Freedy et al (1976)** Michie et al (1968)** Rapp (1972)* Smith, H.T. & Crabtree (1975)** Yntema & Clem (1965)*
Alternative Selection and Execution	The last phase of problem solving is concerned with implementing the solution.	Automatic execution of user-specified actions can aid the user in interacting with the problem-solving environment. Aids that automatically take over the problem-solving process may also be useful, but they should be used with caution.	Bursky et al (1968) Freedy et al (1976)** Hanes & Gebhard (1976)* Pulfer (1971)

Table A-9

Types of Problem-Solving Aids

Aiding Mechanism	Description	Comments	Principal References
Alternative Evaluation	These aids may either automate the user's evaluation criteria, require the user to use established criteria, or simulate the results of actions that do not have well established evaluation criteria.	Except for aids that automate the user's evaluation criteria, these aids are task-specific. Most useful if the task is not well-defined or if a large number of evaluation criteria need be considered.	Brown et al (1975) Hormann (1967) Rapp (1972)* Smith, H. T., & Crabtree (1975)**
Alternative Generation	These aids are primarily used to generate alternatives that the user would not normally consider or, for extremely well-defined tasks, to present algorithmically determined alternatives.	Except for well-defined task domains, where they may have very little impact, they are difficult to construct. Can be cost-effective for training applications, but generally are of limited use in complex problem-solving tasks.	Baldwin & Siklossy (1977) Gagliardi et al (1965)**
Automatic Action Execution	Such aids permit the user to name the desired action without explicitly carrying out the steps involved in its execution.	Most useful when the results of applying an action do not impact subsequent problem-solving actions. If this is the case, the user may need sophisticated alternative evaluation heuristics.	Carlson & Hodgson (1977) Hanes & Gebhard (1966)* Pulfer (1971)
Automatic Takeover	This type of aid functions as an automated decision maker that is able to select alternative actions on the basis of prior observations of the human decision maker's behavior. Although allocation of control to this aid occurs automatically, whenever some criterion of correspondence between predicted and observed human behavior is reached, voluntary turnover of control is also possible.	Although demonstrated to be effective in some contexts (e.g., control tasks), the range of tasks in which this is appropriate is not well understood. User acceptance may be low and should be carefully examined.	Freedy et al (1972) Steeb & Freedy (1976)*
Backtracking	Such an aid allows the problem solver to "undo" the effects of recent actions and return to an earlier state of the problem-solving process without actually starting over.	Useful in tasks where it is possible to "undo" recent actions. Can improve performance at relatively little development cost.	Carlson & Hodgson (1977) Michie et al (1968)* Teitelman (1972)*
Better Weighting of Unreliable Data	This aid re-codes low-fidelity data into a form that is more readily useable by the problem solver.	Depends on the ability to accurately recode low-fidelity data.	Topmiller (1968)** Howell & Gettys (1968)*
Change of Problem Representation	Typical implementations of this aid present problems as isomorphic variations of more standard problem representations. It is intended that this will aid the problem solver in selecting an appropriate and efficient problem formulation.	Most useful in well-understood tasks. An inappropriate representation may seriously degrade performance.	Chesler & Turn (1967) Smith, H. T. (1974)** Newsted & Wynne (1976)*

Table A-9 (Concluded)

Types of Problem-Solving Aids

Aiding Mechanism	Description	Comments	Principal References
Decision Consistency Improvement	This type of aid assists the users applying their own decision strategies consistently in cases in which these strategies are complex.	Useful for expert problem solvers in well-defined tasks. Including sufficient versatility to adapt to individual users may be difficult.	Davis et al (1975)** Freedy et al (1976)**
Decision Strategy Improvement	Such aids assist the user in applying problem-solving techniques that would not normally be considered or known.	Useful in well-defined tasks in which optimal, or near optimal, problem-solving techniques are known, or in tasks in which general heuristics, such as problem reduction, are applicable. Requires detailed knowledge of the task.	Caruso (1970)* Gagliardi et al (1965)** Rogers et al (1964) Wilde (1969)*
Decomposition and Recombination	This type of aid allows the user to divide the original problem into subproblems. The solutions of the various subproblems are then combined into a solution to the original, larger problem.	Useful only if a task can be decomposed into independent subproblems. Requires a good understanding of the task.	Krolak (1971)*
Disruption of Psychological Set	Such an aid is intended to disrupt to any bias or "sets" that the user may employ and, thereby, stimulate more creative, or novel, problem-solving attempts.	Potentially useful, but may disrupt an appropriate "set".	Stewart (1976)
Extended Memory	This aid allows the user to store and retrieve problem-relevant information. This information may initially be generated by the user or by other problem-solving aids, such as aids for alternative generation and evaluation.	Very useful in almost all tasks. Success is related to the ease of retrieval from external memory.	Balzer & Shirey (1968) Newsted & Wynne (1976)* Smith, H. T., & Crabtree (1975)**
Lockout	In an interactive problem-solving situation, this technique restricts the problems solver's access to the computer for some period of time after the presentation of the results from the current request for information.	Although demonstrated effective in some contexts, user acceptance was low. The tradeoff between user performance and user acceptance should be carefully considered.	Buchan et al (1971)** Seven et al (1971)**
Rapid Trial-and Error	This aid allows the user to rapidly and easily examine the consequences of alternative action by simulating their application.	Easily implemented in well-defined tasks. May offset inadequacies in decision strategy improvement aids.	Balzer & Shirey (1968) Carlson & Hodgson (1977) Rapp (1972)* Wilde (1969)*
Strategy Capture	These aids attempt to model and predict the user's behavior. Strategy capture is generally used in conjunction with other aids, such as automatic takeover or alternative evaluation.	A prerequisite for developing automatic takeover aids. Best suited to tasks that allow algorithmic, rather than heuristic, strategies.	Doutriaux (1973)* Freedy et al (1976)**

Table A-10

Some Major Properties of Interactive Dialogues

Dialogue Property	Description	Comments	Principal References
Initiative	Initiative is concerned with whether the user or the computer initiates the individual information transactions within the dialogue. If the computer asks questions, presents alternatives, etc., and the user responds, the dialogue is "computer-initiated". If the user inputs commands without such computer "prompting", the dialogue is "user-initiated". "Mixed initiative" and "variable-initiative" dialogues are also possible.	Computer-initiated dialogues are preferrable for naive users or trainees, and for casual users. Computer-initiated dialogue allows reliance on passive, rather than active, vocabulary, implicitly teaches the user a "system model", and allows use of the system by a user who has not yet internalized such a model. Computer-initiated dialogue is also satisfactory for experienced users if use involves few transactions or system response is very fast. The latter usually implies a "smart" terminal. A slow, computer-initiated dialogue is very disruptive to the frequent, experienced user. See the later section on "response time". For most systems, designers should consider allowing the user to select either dialogue mode.	Johnson (1977) Martin (1973) Thompson (1969, 1971)
Flexibility	Flexibility is a measure of the number of ways in which a user can accomplish a given function. *High flexibility* can be achieved by providing a large number of commands, by allowing the user to define or redefine commands, etc.	There is evidence that nonprogrammer users adopt a "satisficing" strategy with respect to flexible dialogues. That is, they tend to utilize known methods for solving a problem even when the system provides less cumbersome methods, known but not yet learned by the users. There is also evidence that more flexible dialogues degrade performance (especially by increasing error rates) of relatively inexperienced users. Thus, highly flexible dialogues may be undesirable except for experienced and/or sophisticated users.	Eason (1976) Stewart (1976b)* Walther (1973)** Walther & O'Neil (1974)**
Complexity	Complexity is related to flexibility. Complexity is a measure of the number of options available to the user at a given point in the dialogue. Low complexity can be achieved by using few commands, or by partitioning the commands so that the user selects from a small set at any given time.	The effects of dialogue complexity on performance are unclear. It seems reasonable to expect that there is some optimal level of complexity for a particular task and user type, with degraded performance resulting from significantly more or less complex dialogue structure. There is evidence which suggests that a large number of redundant or irrelevant commands impairs user performance, and that extreme "simplification" of the dialogue by hierarchic structuring is also detrimental. Complexity appears to interact with user properties, but a systematic description of this interaction is not yet possible.	Baker & Goldstein (1966)** Botes (1977) Carlisle (1974)**

Table A-10 (Concluded)

Some Major Properties of Interactive Dialogues

Dialogue Property	Description	Comments	Principal References
Power	Power is the amount of work accomplished by the system in response to a single user command. In a dialogue with powerful commands, the user may accomplish with a single command, an operation which would require several commands in a system with less powerful commands. Power is related to flexibility and complexity.	Obviously, the power of the commands must somewhat correspond to the user's needs, which may vary over time and users. In some application areas, a large range of command power is possible and use of relatively high-power commands can be very effective (e.g., matrix operators in a mathematical system). If power is achieved by using powerful commands and facilities instead of less powerful, but more basic commands, the result is a reduction in generality of the system. The MICA study found this to be a significant factor in system rejection by both managers and technical personnel. On the other hand, provision of powerful commands in addition to a more basic set tends to increase dialogue complexity. One possible solution is to partition the dialogue so the less sophisticated user is exposed to a subset of commands.	Eason (1976)* Stewart (1976b)* Wood (1972)
Information load	Information load is a measure of the degree to which the interaction absorbs the memory and/or processing resources of the user.	In most tasks, user performance is adversely affected by information loads which are either too high or too low. Obviously, information load is a function of the task and the user's familiarity with the task, but it is also affected by the design of the dialogue. The information load associated with a task can be measured empirically or estimated, and can be altered by changing display mode and/or channel capacity, power of commands, use of default values, command language type and structure, and a variety of other dialogue details.	Finkleman (1976) Meister (1976) Parsons (1972) Treu (1975)
System response time		See next two figures.	
Communication medium		See later sections dealing with natural-language dialogue, interactive graphics, input device selection and output device selection.	

Table A-11

Summary of System Response Time Effects

Factor	Comments	Principal References
System response time	If below user preparation time, involves little or no performance advantage, and may, in certain cases, actually degrade user performance	Franklin & Dean (1974)* Newman (1969)**
	Otherwise, delay in excess of that tolerable for specific task results in step decrements in performance and results in user dissatisfaction.	Miller (1968) Morefield et al (1969)**
	With excessive delay, user may adapt mode of system use by time-sharing own time or by a variety of partial or total withdrawal strategies	Carbonell et al (1968) Eason (1976)
System response time variation	If significant, may result in user dissatisfaction. Performance effects are probably mostly related to individual SRTs, as discussed above.	Carbonell et al (1968) Simon (1966) Williams (1975)**
Display writing time	Has been subjected to limited study as separate issue. No clear conclusions.	
Artificial lockout	May improve performance in complex problem solving. Reduces user satisfaction.	Boehm et al (1971)** Gold (1967) Seven et al (1971)**

Table A-12

Basic Interactive Dialogue Types

Dialogue Type	Description	Comments	Principal References
Question-and-Answer	Computer asks a series of questions, to which user responds.	Inherently computer-initiated. For totally naive user, this is probably the most error-free dialogue type. This approach rapidly becomes cumbersome as the user gains experience	Card et al (1974)** Lucas (1977)**
Form-filling	Computer presents form with blanks. User fills in blanks	Computer-initiated. Faster than ordinary question-and-answer dialogue, because user provides several responses in a single transaction. When user input is dominated by parameter values, rather than commands, this approach is often best. Other than casual use requires terminal with tabbing feature. Under some circumstances, a significant proportion of syntactic data entry errors may be detected if terminal has provision for imposing constraints on data by field.	Martin (1973) Pew & Rollins (1975) Straub (1975)**
Menu Selection	Computer presents list of alternatives, and user selects one or more	Inherently computer-initiated. Can be used for command construction as well as data base search. Very natural dialogue if response-time criteria are satisfied and "point-in" selection device (e.g., light pen, touch panel) is used.	Martin (1973) Ridsdale (1970)* Thompson (1969, 1971) Uber et al (1968)
Function Keys with Command Language	User indicates desired action by depressing keys, each of which represents a command, command modifier, or parameter value	User initiated, but with keyboard as memory aid. Can be computer-initiated if "programmable" keyboard or tutorial displays are used. Often appropriate when user input is dominated by commands, rather than parameter values. Appropriate for naive user only if command syntax is very simple and/or computer-initiated form is used; otherwise requires training. Constant presence of all commands and modifiers may make it difficult for user to learn appropriate model (e.g., hierarchical structure) of command language, if language is not simple	Martin (1973)
User-Initiated Command Language	User types commands, perhaps using mnemonic abbreviations	Acceptable approach for well-trained user who has fully internalized model of system function and language syntax. Otherwise error-prone and sometimes frustrating. Usually preferred by system designers and programmers, who tend to satisfy these criteria often applied uncritically by them to systems in which user does not satisfy criteria.	Martin (1973)
Query Language	User inputs questions or data-base access procedures to a data base system. System produces response or report	Several existing query languages appear to be useable by both novices and programmers, but many errors occur in their use. The problem areas appear to be reasonably well known, but only fairly general solutions can be confidently prescribed. Detailed guidelines for query language design would be premature.	Gould & Ascher (1975)** Reisner (1977)** Thomas (1976)* Thomas & Gould (1975)**
Natural Language	Dialogue is conducted in user's natural language (e.g., English)	Can be user- or computer-initiated or mixed initiative. Fairly high-powered natural language capabilities are now achievable. Cost is very high, however, since the system requires an extensive "knowledge" of the application area in order to understand user inputs. Development of such a data base is definitely a nontrivial task, and is not for unsophisticated designers. "Natural" language may not be the most "natural" dialogue type for many applications (e.g., engineering drawing, mathematics).	See text discussion
Interactive Graphics	Generation of pictorial displays, ability of user to select displayed entities and spatial locations by pointing or similar nonverbal means	This is not truly a dialogue type, but interactive graphics offers great additional dialogue flexibility. With rapid response (as by use of smart terminal with refreshed graphical CRT display), very rapid, flexible and "natural" dialogues are possible. Interactive graphics is relatively expensive, but costs are dropping. Performance improvements may offset extra cost even in some relatively mundane applications (e.g., search of hierarchical data base). Most research on interactive graphics is concerned with details of input and output devices and techniques, rather than overall dialogue properties. See later sections on those topics.	Foley & Wallace (1974) Martin (1973)

Table A-13

Some Known Problem Areas in the Use of Query Languages

Problem Area	Comments	Principal References
Logical Quantifiers	Use of logical quantifiers (all, some, none) in the presence of set relations (union, intersection, etc.) is very error-prone.	Thomas (1976)**
Set Relations	When sets of elements are related in a complex way, human interpretation of the relationship is often erroneous.	Thomas (1976)**
Logical Relations	Disjunction (logical "or") and negation are error-prone constructs. Although this finding is consistent with basic psychological research and is probably generally true, the principal study (Miller, 1974) in which it is related to programming and query language design involves an typical task in which the subject must compensate for absence of these constructs in a language by devising a procedural specification using transfer of control.	Gould & Ascher (1975)**
Arithmetic Relations	Conversion of inequalities (e.g., from "over 50" years old to "51 or more" years old) is error-prone. Also, users tend to use arithmetic relations even with "nominal" categories, as "college degree greater than or equal to B.S."	Gould & Ascher (1975)**
Semantic Confusion of Commands	Errors occur when query language has confusable commands, such as COUNT ("how many numbers are there?") and TOTAL ("what is their sum?").	Gould & Ascher (1975)**
Use of Synonyms for File Names, Properties, etc.	Users tend to substitute synonymous terms (e.g., "employee" for "personnel" file) which system may not recognize.	Reisner (1977)**
Misspelling	Spelling errors are common in query formulation. Also, users tend to use an incorrect ending (e.g., "employees" instead of "employee").	Gould & Ascher (1975)** Reisner (1977)** Thomas & Gould (1975)**
Omission of Problem-Relevant Attributes	In formulating complex queries, users frequently omit one or more of the attributes which define the set.	Gould & Ascher (1975)** Thomas & Gould (1975)**
Contextual Referencing	If unconstrained, users tend to make contextual references in queries. However, there is no clear evidence that users fail to adapt to query languages which preclude such references.	Miller & Becker (1974)**

Table A-14

Basic Display Device Types

Display Type	Comments	Principal References
Refreshed CRT	The ordinary, refreshed CRT is currently the "basic" computer display. A good deal of data exists concerning appropriate visual properties of CRT displays. See later discussion of Display Properties. Studies which have compared user performance using CRTs with performance on other display devices do not provide a satisfactory basis for selection decisions (see text).	
Storage Tube CRT	For some graphical applications, direct-view storage tubes may be preferable to refreshed displays. The storage tube allows very high-density, flicker-free displays, but imposes significant constraints on interactive dialogue. Although information exists concerning the basic functional advantages and disadvantages of such displays, no empirical data pertaining to human factors concerns were found.	Steele (1971)
Plasma panel display	Plasma panel displays are inherently "dot", or punctate, displays, and studies of symbol generation method are relevant (see later section on Display Properties). Little empirical information exists on human performance aspects of plasma displays per se.	
Teletypewriter	Reasonable guidelines exist with respect to the design of teletypewriter terminals, including both physical and functional properties. See later discussion of Terminals.	Dolotta (1970)
Line printer	Research on typography is voluminous and directly applicable. Research dealing directly with the line printer used in computer output is scanty, but consistent with findings of typographic research (e.g., mixed upper-lower case is best for reading comprehension). Guidelines are not known to exist, but could be constructed with additional survey of typographic research literature. Use of line printers for "pseudographic" displays is common, little discussed in the literature. Pseudographics is an inexpensive way to convey simple graphical information, and should probably be used more widely in batch applications.	Lewis (1972)** Ling (1973) Poulton & Brown (1968)**

Table A-14 (Concluded)

Basic Display Device Types

Display Type	Comments	Principal References
Laser displays	Reasonable human factors guidelines with respect to visual properties have been proposed, but these displays are not widely used.	Gould & Makous (1968)
Tactile displays	Although some tactile displays have been proposed or even developed, little human factors research has been done other than that concerned with prosthetics.	Noll (1972)
Psychophysiological displays	Psychophysiological input is technically feasible now, but psychophysiological displays are still only a topic for research.	
Large-screen displays	There is conflicting evidence with respect to the performance effects of large-group vs individual displays. The main advantages of large-screen displays are a larger display area and the existence of a single display which is clearly the same for all viewers. Unfortunately, higher display content is not achievable due to the resolution limits of existing technology (e.g., light valve displays), and may be unachievable in principle, since the large-screen display usually subtends a smaller visual angle than an individual display located close to the user.	Landis et al (1967)** Smith & Duggar (1965)**
Telephone	See later discussion of Terminals.	
Graphical displays	See later discussion of Graphical Displays.	

Table A-15

Basic Visual Properties of Displays

Property	Comments	Principal References
Flicker	Perceptible display flicker can cause irritation and visual fatigue, and may impair visual performance. Flicker is a function of regeneration rate, phosphor persistence and chromaticity, and size of display element. Beam scanning sequence (as in interlaced raster scanning) can affect the disturbing effect of flicker, but has only a small effect on perceptibility of flicker. Good guidelines and evaluation methodology exist.	Dill & Gould (1970)** Gould (1968)*
Luminance, Luminance Contrast	Appropriate levels of luminance and liminance contrast may be affected by room illumination and display chrominance. Good guidelines exist.	Gould (1968)*
Chromaticity	Chromaticity of single-color displays is probably not a major issue. The yellow-green portion of the spectrum corresponds to the greatest sensitivity of the eye, but adequate luminance is the issue there. Possible effects of chromaticity on eyestrain is an open issue. See discussion under "color coding" for information pertaining to multicolor displays.	Gould (1968)* Mezrich et al (1977)**
Resolution	Some guidelines exist with respect to display resolution, but they vary with visual task, type of display element, display element size, etc. For alphanumeric characters, raster displays should provide at least 8 and perhaps 10 scan lines per character height, although the required number varies inversely with character height over a significant range. Dot matrix displays should use at least a 5 x 7 matrix, and preferably 7 x 9. Construction of more extensive guidelines considering visual task, etc., is possible, but would turn up numerous gaps. Good techniques exist for evaluating legibility of specific display.	Buckler (1977) Gardner & Soliday (1974)** Hemingway & Erickson (1969)** Smith & Goodwin (1973)** Vartabedian (1970a)**

Table A-16

Display Properties of Alphanumeric Characters

Property	Comments	Principal References
Generation method	For visual search, and perhaps reading, dot-matrix characters appear to be slightly preferable to stroke-generated characters. Resolution and font are related issues. In particular, if individual dots are easily perceptible, small area flicker can occur, and there is evidence that characters composed of perceptible dots may be encoded in human memory in a special way. This might affect performance on matching tasks, for example.	Buckler (1977) Maddox (1977)* Peters & Barbato (1976)** Vartabedian (1971)**
Font	Simple character fonts are best, with no serifs, variable stroke width, slanting, etc. Dot-matrix characters using a 7 x 9 dot matrix yield better performance than with 5 x 7 format, but 5 x 7 appears satisfactory for most tasks. The standard Leroy font (and similar MIL-M-18012) appear satisfactory for computer displays, as well as hardcopy.	Buckler (1977) Vartabedian (1970a)**
Size	Desirable character size (measured in angular subtense) varies with visual task (e.g., larger characters are required for single character search and recognition than when whole words are involved). Required character size varies inversely (within a limited range) with display resolution, at least in raster displays. Reasonable guidelines exist.	Buckler (1977) Giddings (1972)** Hemingway & Erickson (1969)** Whitham (1965)
Case	Evidence seems to indicate that the use of all upper-case characters is preferable for visual search tasks involving whole words, but that ordinary mixed upper and lower case is better for reading tasks.	Poulton & Brown (1968)** Vartabedian (1971)**
Spacing	Data from non-electronic displays suggest that intercharacter spacing should be 26-63% of character width. Virtually all electronic displays use uniform spacing, rather than the proportional spacing used in typesetting, but no data were found indicating the performance effects of this practice.	Buckler (1977)
Aspect ratio	Symbol height-to-width ratio does not appear to be a significant parameter. A range of 1:1 to 2:1 seems acceptable. On non-electronic displays, a ratio of 4:3 has been found most legible.	Buckler (1977)
Cursor	On an alphanumeric display, the cursor marks the character position at which changes, insertions, etc., will occur. Several cursor forms seem to be acceptable based on visual search and cursor placement times, and user preference ratings. Acceptable types include a box, diamond, cross, or underline blinking at 2-5 Hz. The cursor should not blank out or modify the character it marks. Nonblinking cursors yield inferior search performance. Best overall performance, of those tested, was a box blinking at 3-4 Hz.	Vartabedian (1970)**

Table A-17

Major display coding techniques

Coding Method	Comments	Principal References
Alphanumeric coding	Alphanumeric coding is best for absolute identification, but you may involve problems of size (coding usually is accomplished by adding additional symbols, which take up space on the display), confusability of similar symbols, learning of associated meanings, and sometimes superimposition of coded symbols.	Christ (1975)* Grether & Baker (1972)
Shape coding	Shape coding is useable for both search and identification tasks. The use of meaningful shapes or geometric symbols is widespread, but has received little empirical study. A key issue is symbol discriminability.	Christ (1975)* Grether & Baker (1972)
Color coding	Relevant color coding -- redundant or nonredundant -- generally yields better performance than other coding methods in both search and identification tasks, except that alphanumeric coding yields better identification accuracy. There are minor exceptions, however, and performance advantages of color are much greater in some situations than in others. There are a number of other factors which might influence the selection of color as a coding mechanism. Users tend to prefer color even when there is no performance advantage, and possibly when the overall effect on performance is negative.	Christ (1975)* Teichner et al (1977)**
Blink coding	Blink coding is effective as an inclusion or exclusion code for target detection tasks, but can adversely affect reading performance if user cannot match scan rate to blink rate. Blink coding helps most with high density displays. Although users can discriminate up to four blink rates, blink coding should probably be restricted to a binary code (1 class flashing, 1 static). In this case, optimal flash rate is probably 3-4 Hz. No research is known on optimal on-off cycle.	Smith & Goodwin (1971, 1972)**
Others	See text.	

Table A-18

Input Device Types

Input Device	Comments	Principal References
Keyboard	The vast majority of past research on input devices has dealt with keyboards. Reasonable and fairly detailed guidelines exist with respect to the physical properties of keys and keyboards and -- to a lesser extent -- their layout, logical properties, operating procedures, etc. Guidelines for alphabetic keyboards are particularly good, and those for numeric keypads are reasonable. Function keyboards are rather system-dependent; guidelines can specify their physical properties, but can only suggest methods and basic principles for function selection and layout. It is not clear that chorded keyboards are viable except in highly specialized situations.	Alden et al (1972) Seibel (1972)
Lightpen, lightgun	Lightpens can be used effectively for cursor placement and text selection, command construction, and for interactive graphical dialogues in general, including drawing. However, there is evidence that greater accuracy may be possible with a mouse in discrete tasks, and with a trackball in drawing tasks. Mode mixing, as by alternating use of lightpen and keyboard, can significantly disrupt performance, since the lightpen must be picked up and replaced with each interval of use. Continuous use of a lightpen, at least on commercially available CRT terminals with vertical display surfaces, can be quite fatiguing. There has been no known research on desirable physical and logical properties for lightpens.	English et al (1967)** Goodwin (1975)** Irving et al (1976)** Ramsey (unpublished)**
Joystick	There are many studies of the use of joysticks for continuous tracking tasks, but few studies of its use for discrete or continuous operand selection or graphical input tasks. Those studies which have been performed have found the mouse, lightpen, and trackball preferable in terms of speed, accuracy, or both. Joysticks are sometimes used for windowing and zooming control in graphical displays. No research on this topic was found, although the results of tracking studies may be applicable here. Otherwise, no clear recommendations for joystick properties emerged from the survey, even with respect to basic issues like position vs. rate vs. acceleration control. These issues may be fairly task-specific.	Card et al (1977)** English et al (1967)** Irving et al (1976)**
Trackball	The trackball appears to be effective for both discrete and continuous operand selection and graphical input tasks, and may yield the best performance when graphical inputs must be alternated with keyboard input. No empirical data on physical properties were found, but some such data are thought to exist in the tracking literature.	Irving et al (1976)**

Table A-18 (Continued)

Input Device Types

Input Device	Comments	Principal References
Mouse	Although the mouse is not in widespread use, there is evidence that it is an effective device for text selection. No data are known concerning its physical properties, or its use in other tasks.	Card et al (1977)** Engelbart (1973) English et al (1967)**
Graphical input tablet	Graphical input tablets are capable of fairly high pointing accuracy (within 0.08 cm, according to one study). They are commonly used for freehand drawing, but may be inferior for discrete position input tasks. They may also involve a performance decrement due to low stimulus-response compatibility when the drawing surface is separate from the display surface.	English et al(1967)** Myer (1968)**
Touch panel	No empirical performance data were found dealing with the touch panel. While its inherent resolution limits may preclude serious use for fine discrete position and continuous position input, it feels "natural" and may become a common device for more coarse positioning and selection from lists.	Hlady (1969) Johnson (1977)
Knee control	A knee control has been used in one research study for discrete position input. It is not known to be in use otherwise, and seems unlikely to see serious use.	English et al (1967)**
Thumbwheels, switches, potentiometers	These have been studied primarily outside the computer systems domain, and are discussed in standard human factors reference sources. They are not often used as major user input devices for interactive computer systems.	
Tactile input devices	Although some tactile input devices have been proposed, little human factors research has been done other than that concerned with prosthetics.	Noll (1972)
Psychophysiological input devices	Electromyographic signals have provided superior performance in some control tasks to joysticks and other manual control devices. Use of heart rate, keyboard response latency, EEG input, etc., is technologically feasible, although really sophisticated input is not yet achievable via these methods. There are ethical and legal problems here, as well as technological difficulties. Significant human factors data were not found with respect to computer-related use of these techniques.	Slack (1971) Wargo et al (1967)**

Table A-18 (Concluded)

Input Device Types

Input Devices	Comments	Principal References
Automated speech recognition	The current state of this technology limits its use to relatively simple input tasks. Even there, there are problems with different speakers, noise, etc. Although speech input seems like a very desirable and natural input mode, and is clearly preferred over other communication modes for interpersonal communication, it is not clear whether it will prove to be widely applicable for human-computer interaction tasks. Very little information was found which would assist the designer in recognizing tasks for which speech input is appropriate, or in selecting an appropriate speech input device.	Addis (1972)* Bezdel (1970)** Braunstein & Anderson (1959)** Chapanis (1975, 1976)* Turn (1974)
Hand printing for optical character recognition (or for subsequent entry by typist)	The constrained hand printing required for OCR input results in low input rates, and sometimes high recognition error rates as well. Although manual transcription of such data clearly cannot be avoided in many cases, the preponderance of evidence suggests that direct keyboard entry yields better performance than printing, with a little practice, even when users are not skilled typists. Some error and input rate data on handprinting exist, along with some information about the effect of various printing constraints on input performance.	Apsey (1976)** Devoe (1967)** Masterson & Hirsch (1962)** Smith (1967)** Strub (1971)**
Mark sensing	As with hand printing, this separate transcription results in lower input rates than does practiced, but unskilled, typing. Some error and input rate data exist. May be slightly faster than constrained hand printing.	Devoe (1967)** Kulp & Kulp (1972)**
Punched cards	Keypunching performance differs significantly from ordinary typing because of differences in both the machine and the typical data to be keyed. Some reasonably good data exist on keypunch timing and error rates.	Neal (1977)**
Touch-tone telephone	Several studies suggest that the touch-tone telephone is a satisfactory device for occasional use as a computer terminal, even by naive computer users. It seems clear, though, that it is not a satisfactory device for prolonged interaction or for significant amounts of non-numeric input.	Miller (1974)** Smith & Goodwin (1970) Witten & Madams (1977)

APPENDIX B

MMI REQUIREMENTS MATRIX

In Section 2 of this report, there is a discussion of the possibility of developing a structured approach to the definition of man-machine interface requirements. More specifically, it is proposed to develop a matrix in which MMI functional capabilities are listed as row labels, identified operator tasks are represented as columns, and cell entries estimate the degree to which each capability is required for each task. A first pass at such a requirements matrix is presented here in Table B-1. Some further discussion is needed concerning the listing of MMI functions, the characteristic tasks chosen for illustration here, the estimation of requirements, and the potential application and extension of this requirements matrix.

MMI FUNCTIONS

A first pass at listing MMI functional capabilities is represented by the several hundred row labels in the requirements matrix. This tabulation has been drawn from several sources, but particularly from the listing of MMI features by Goodwin (1978). Capabilities are expressed in functional terms, for the most part, without reference to specific means of implementation, e.g., "position designation" rather than "lightpen".

MMI functions are grouped here under six major headings, generally following their discussion in the text of this report. First, and most fundamental, is Dialogue Type. Specification of dialogue type will influence the definition of all other MMI functional requirements.

Other functional areas listed here include Data Entry, Data Display, Sequence Control (strongly related to Dialogue Type) and User Guidance. An additional area, Data Transmission, is listed to indicate that this requirements matrix can be extended to cover a broader scope of MMI functions than those that have been discussed in this report. Data Transmission includes consideration of automatic data inputs from external systems, which may affect MMI functional design, as well as data transfers resulting from explicit operator actions.

This listing of MMI functions needs further extension to include capabilities not considered here, e.g., those relating more

specifically to maintenance tasks, system performance evaluation and testing.

This listing of MMI functions also needs amplification in some areas, based on cumulative experience in using the requirements matrix. Certainly the specification of various dialogue types (codes 1.x) could be amplified to include sub-headings, perhaps including response time requirements (as suggested in Table 3-1) among other things.

Rows now labeled "other" (codes 2.1.2.2, 2.1.3.3.4, 2.2.2, etc.) could be amplified to cite more specific alternatives. Rows labeled "automatic" (codes 2.6.1.1, 2.6.3.1, etc.) could be amplified to distinguish functions driven by "external" events from those resulting implicitly from operator inputs and control actions.

Some of the functions listed here, although seemingly specific, may on closer examination be defined more effectively in terms of component sub-functions. For example, text editing (code 2.3.1.2.2) could be elaborated in terms of subfunctions to type, insert or delete characters, to mark character strings, to move, copy or delete marked strings, and to save text in named files, or named or unnamed buffer stores (Goodwin, 1978, p. 55).

Aside from the need for extension and amplification, this listing of MMI functional capabilities should be examined critically as to the proper placement or grouping of functions. In practice, the close association of Sequence Control with Dialogue Type, noted above, may make it unfeasible to distinguish these as separate topics in requirements definition. A less significant example has to do with the questionable placement of alarm functions, which are listed here under Sequence Control (code 4.5) to emphasize potential user participation in determining alarm logic, but must also be acknowledged under User Guidance (codes 5.1.5, 5.3.3) which is the general purpose of alarms, and considered in connection with other more specific functions (see code 6.3.4.1).

CHARACTERISTIC TASKS

To illustrate the potential application of this requirements matrix, four characteristic data processing tasks are shown in Table B-1 as columns in the matrix. These are not "generic" tasks, as that word is used in the text of this report (see page 16), but they should serve to illustrate characteristic patterns of functional requirements. These four are, of course, only a small sampling of the broad variety of tasks that must be performed in C3 systems. In the subject index to a recent bibliography on man-

computer systems (Ramsey, Atwood and Kirshbaum, 1978), 34 general application areas are cited, from air traffic control to wargaming. The variety of different possible operator tasks is probably at least as large, although that could only be verified by comparative examination and categorization of task analysis descriptions.

The four tasks tabulated here, although a small sample, do represent a rather broad range of operational jobs. Two represent "operator" tasks, the third is a poorly-defined "analyst" task, and the fourth a narrowly-defined "data entry/service" task, as those terms have been applied to characterize user roles (Goodwin, 1978, pages 43-44; see also text, pages 15-16).

The first task is labeled Mission Scheduling. This is the kind of data handling task that must be accomplished, for example, at a Tactical Unit Operations Center (TUOC), assigning sorties from several squadrons of aircraft to a list of preplanned missions, trying to optimize aircraft, armament and crew capabilities in relation to target characteristics, trying to minimize flight distances and on route refueling requirements, trying to schedule takeoffs to meet required time over target and schedule returns to provide time for necessary turn around. TUOC mission scheduling, as conceived here, requires rather little data entry: mission and target data are filed automatically with receipt of orders from the Tactical Air Control Center, and current squadron status data are assumed already available. The task itself is well defined. The operator makes iterative matchings of aircraft and crews against missions, sometimes changing those that do not fit well in the developing schedule. Computational aids are needed for time/distance calculation and would be helpful in assessing proposed alternative plans. The final schedule must be reported to higher command and to squadron operations officers, and also filed for subsequent operations monitoring. Scheduling of preplanned missions can be done under relatively little time pressure. The operator can probably accomplish most of the required transactions simply by pointing at categorized displays, so that an extended form of menu selection is the preferred dialogue type.

The second task is labeled Operations Monitoring and Control. This is the kind of task that might be required, for example, in the Communication System Control Element (CSCE) proposed for TRI-TAC acquisition of tactical communication control facilities (see Smith, 1974). In this task the operator is expected to monitor on a continuous basis the flow of message traffic through the nodes and channels of a communications network, to spot overloads and outages, and to institute control actions as necessary to maintain network functioning. The operator may have to work under time pressure,

assimilating information from rapidly changing network displays, so that some form of auxiliary display coding will probably prove valuable. Computational aids could help the operator predict the result of proposed network reconfiguration. Communication functions are needed to transmit control messages to node elements in the network. Rather little data entry is required in the performance of this task, and menu selection combined with the direct designation of nodes and links on a graphic display is probably the best dialogue type to use.

The third task is labeled Intelligence Analysis. This task is generally conceived to be like those proposed for the Department of Defense Intelligence Information System (DODIIS), where an analyst must access an extensive data base in a variety of ways in order to answer a broad range of questions. Some form of query language is the dialogue type required here. Other functional capabilities include flexible formatting of display output, and computational aids to permit categorization, statistical summary and inference from stored data. The analyst would probably be a periodic rather than a continuous user of the computer facility, will draw on other sources of information, will need capabilities for entry and manipulation of a variety of data types, including some text processing and graphics.

The fourth task involves routine Data Entry of the sort required when loading or unloading air cargo (see Smith, 1976b). The task must be performed under time pressure by relatively unskilled personnel. Data formats and the sequence of entries are rigidly controlled. A variety of data validation checks are applied to ensure correct entry and proper cargo handling. This task is probably best designed as keyed data entry with prompting, in a question-and-answer type of dialogue.

More tasks should be added as the requirements matrix gains acceptance and practical use. The eventual list should include maintenance tasks, with their emphasis on diagnostic routines and performance evaluation, as well as the kinds of operational tasks illustrated here.

REQUIREMENT ESTIMATES

MMI requirements are represented by the cell entries in the matrix. Here only rough estimates are given. For each task every functional capability has been judged as Essential, Useful or Not Needed, a three-way categorization suggested in the text of this report (see page 23).

Where a function is marked Not Needed, any sub-functions under that heading have simply been marked with a dash to indicate that no separate judgments had to be made.

The logical constraints relating ratings for functions and sub-functions also work in reverse. In this requirements matrix, each function has been assigned a rating corresponding to the highest rating given any of its related sub-functions.

If requirement estimates for some functions are redundant, i.e., directly inferrable from ratings of sub-functions, then why include them in the matrix? Experience in applying the requirements matrix may in fact confirm that redundant ratings can be omitted. They should be retained, however, if they facilitate scanning the matrix, or the association of design guidelines with required functions, or the possible specification of MMI requirements on the basis of higher-level functions rather than detailed sub-functions.

In the text of this report it is suggested that future experience may permit replacing qualitative requirement estimates in the matrix with quantitative indices. This has not been done here, and some awkward expedients have been adopted to deal with certain necessarily quantitative capabilities. An example is the estimation of required display capacity (see entries under code 3.2). Rather than simply indicating anticipated capacity requirements as cell entries in the matrix, a categorization of high-moderate-low has been incorporated in the list of functional capabilities. Such a "solution" enlarges the requirements matrix to little purpose and so does not seem satisfactory.

In actual application of the requirements matrix to MMI functional specification, it may prove helpful to annotate the ratings of different capabilities, to indicate just how those judgments were reached. Supplementary annotation would certainly be needed to explain ambiguous entries, as when any capability called "other" has been rated Essential. Such annotation would also help to delineate the conceptual design of the more general information system of which the MMI will be a critical part.

Even in its present rough form, however, with qualitative entries and no annotation, the matrix does seem to provide a useful structure for MMI requirements definition. The sample tasks included here do seem to show different "profiles" of required capabilities which serve to highlight their characteristic functional differences. Future experience can confirm whether such profiled requirements can help guide MMI design.

APPLICATION AND EXTENSION

Design should follow function. To state function is to imply design. To define functional requirements will necessarily constrain the designer. That is the whole point. Thus requirements must be specified with care, so that arbitrary constraints will not exclude desirable design alternatives. This is particularly true in hardware specification. Constraints in software specification, however, serve generally to make subsequent software design more efficient and ultimately better related to needs. The question is, can use of an MMI requirements matrix constrain design to good effect?

In its present rough form the requirements matrix is little more than a check list. Indeed, if the requirement estimates were simplified to become just a binary categorization of needed versus not needed, then check list would be an apt designation. Even a simple check list, of course, can find useful application, serving to remind system developers of the range of functional capabilities that should be considered, and providing a structure for the imposition of design guidelines.

Use of a check list or simple matrix will not necessarily make requirements definition a simple process. MMI requirement estimation will still involve a good deal of judgment, and if done carefully may imply prior design, at least at a conceptual level, of the overall information system in which the MMI will be implemented.

A further complexity is that most C3 systems involve multiple jobs, each with multiple data handling tasks. Thus ratings and descriptive annotation of the requirements for one task must be combined with those for another task to produce a total MMI functional specification.

As the requirements matrix is extended and amplified from this initial version, its application may become more difficult. The use of quantitative indices in the matrix, as suggested above, could permit a relative weighting and trade-off of requirements, with potential use in evaluating MMI design proposals as well as in early requirements definition. But such elaboration of requirement estimates would probably need computational aids for its effective use. Thus some sort of computer-based version of the requirements matrix would have to be developed.

A further extension of this tool might be to expand the listing of functional capabilities to include other factors influencing MMI design. The present matrix, for example, does not refer directly to

means of implementation. For example, it is never specified whether data displays should be implemented as electronic displays (cathode ray tube or other), on a printer, as an audio output, or by any other particular means. Where a commitment has already been made to particular means of implementation, as in upgrading an existing system retaining operator work station facilities in place, then some way must be found to include such constraints in MMI requirements specification.

Perhaps the best way to handle constraints of this kind -- those relating indirectly to software design, such as committed facilities, unusual work environments, special user characteristics -- is to create a separate check list of expected conditions, rather than to expand the requirements matrix.

A more difficult decision is how to handle constraints that could directly affect software design. In system upgrade, retention of existing software may severely limit options for extending and improving MMI design. Considered more positively, future system acquisition may come to rely on "building blocks" of the kind advocated by Clapp and Hazle (1978), in which successful software implementations of selected MMI functions are preserved as standard modular packages, and translated as necessary for re-use in new system applications.

Once such standard MMI modules have been developed, they should probably be included in any expanded version of the requirements matrix, to remind system planners of their availability and to permit their formal inclusion in system specifications as appropriate. In the long run, increased reliance on such standard components should simplify MMI specification by reducing the number of specific features that are considered in the requirements matrix. But design standardization to that degree still lies many years away.

Meanwhile, it is clear that this initial version of the MMI requirements matrix represents only a partial, tentative listing, based on best judgment rather than proven value. This matrix needs extension and amplification, and its potential usefulness must be evaluated through cumulative experience in trial applications. Suggestions for its improvement are needed and will be welcome.

Table B-1

Sample MMI Requirements Matrix

MMI Capability	Characteristic Tasks* MS	OM	IA	DE
1.0 DIALOGUE TYPE				
1.1 Question and Answer	N	N	N	E
1.2 Form Filling	U	N	U	N
1.3 Menu Selection	E	E	N	N
1.4 Function Keys	U	U	N	N
1.5 Command Language	N	U	U	N
1.6 Query Language	N	N	E	N
1.7 Natural Language	N	N	U	N
1.8 Graphic Interaction	U	U	U	N
2.0 DATA ENTRY/INPUT				
2.1 Position Designation (Cursor Control)	E	E	E	N
2.1.1 arbitrary positions	E	N	U	-
2.1.1.1 discrete	E	-	U	-
2.1.1.2 continuous	U	-	N	-

*MS = Mission Scheduling
OM = Operations Monitoring & Control
IA = Intelligence Analysis
DE = Air Cargo Data Entry

Requirements Ratings:
E = essential
U = useful
N = not needed

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
2.1.2 predefined positions	E	E	E	-
2.1.2.1 "HOME"	N	N	E	-
2.1.2.2 other	E	E	N	-
2.1.3 incremental positions	N	N	U	-
2.1.3.1 by character	-	-	U	-
2.1.3.1.1 right	-	-	U	-
2.1.3.1.2 left	-	-	U	-
2.1.3.1.3 up	-	-	N	-
2.1.3.1.4 down	-	-	N	-
2.1.3.2 by interval ("TAB")	-	-	N	-
2.1.3.2.1 horizontal	-	-	-	-
2.1.3.2.2 vertical	-	-	-	-
2.1.3.3 by other features	-	-	U	-
2.1.3.3.1 "word"	-	-	U	-
2.1.3.3.2 "line"	-	-	U	-
2.1.3.3.3 "paragraph"	-	-	U	-
2.1.3.3.4 other	-	-	U	-
2.2 Direction Designation	N	N	N	N
2.2.1 vector rotation	-	-	-	-
2.2.2 other	-	-	-	-
2.3 Data Type	E	E	E	E
2.3.1 text	N	N	E	N
2.3.1.1 formatted (messages/lists)	-	-	U	-
2.3.1.1.1 entry	-	-	U	-
2.3.1.1.2 change	-	-	U	-
2.3.1.2 unformatted (free text)	-	-	E	-
2.3.1.2.1 entry (composition)	-	-	E	-
2.3.1.2.2 change (editing)	-	-	E	-
2.3.2 tabular	E	E	E	E
2.3.2.1 entry	E	N	E	E
2.3.2.2 change	E	E	E	E
2.3.3 graphic	U	E	E	N
2.3.3.1 entry	U	N	E	-
2.3.3.1.1 selection	N	-	E	-
2.3.3.1.2 creation	U	-	U	-
2.3.3.2 change	U	E	E	-

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
2.4 Entry Formats	E	E	E	E
2.4.1 predefined	E	E	U	E
2.4.2 user defined	N	N	E	N
2.5 Data Validation	E	E	U	E
2.5.1 required entry	E	E	U	E
2.5.1.1 immediate	E	E	N	E
2.5.1.2 deferrable	U	N	U	N
2.5.2 length of entry	E	E	N	E
2.5.2.1 fixed	E	N	-	E
2.5.2.2 maximum	E	E	-	E
2.5.2.3 minimum	E	N	-	E
2.5.3 content of entry	E	E	U	E
2.5.3.1 numeric	E	E	N	E
2.5.3.2 alphabetic	E	E	N	N
2.5.3.3 alphanumeric	E	E	N	N
2.5.3.4 defined codes	E	E	U	E
2.5.3.5 other	N	E	U	N
2.5.4 comparative checks	E	E	U	E
2.5.4.1 equal to	E	N	U	E
2.5.4.2 greater than	E	E	U	E
2.5.4.3 less than	E	E	U	N
2.5.4.4 "IF...THEN"	E	U	U	U
2.5.4.5 other	N	U	U	N
2.5.5 default entry	E	E	N	U
2.5.5.1 predefined	N	N	-	N
2.5.5.2 user defined	E	E	-	U
2.6 Data Processing	E	E	E	E
2.6.1 cross-file update	E	E	U	N
2.6.1.1 automatic	E	E	N	-
2.6.1.2 by request	N	N	U	-
2.6.2 derived data (summary statistics, etc.)	U	E	E	E
2.6.2.1 automatic	U	E	U	E
2.6.2.2 by request	N	U	E	N
2.6.3 other computational aids	E	U	E	N
2.6.3.1 automatic	N	N	N	-
2.6.3.2 by request	E	U	E	-

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
3.0 DATA DISPLAY/OUTPUT				
3.1 Data Type	E	E	E	E
3.1.1 text	N	N	E	N
3.1.1.1 formatted	-	-	U	-
3.1.1.2 unformatted	-	-	E	-
3.1.2 tabular	E	E	U	E
3.1.3 graphic	U	E	E	N
3.1.4 combination	N	E	U	N
3.2 Data Density				
3.2.1 high	U	E	E	N
3.2.1.1 (text > 1000 char./frame)	N	N	E	-
3.2.1.2 (tabular > 600 char./frame)	U	N	U	-
3.2.1.3 (graphic > 300 char./frame)	N	E	U	-
3.2.2 moderate	E	E	E	N
3.2.2.1 (text 600-1000 char.)	N	N	E	-
3.2.2.2 (tabular 300-600 char.)	E	U	U	-
3.2.2.3 (graphic 100-300 char.)	U	E	E	-
3.2.3 low	E	E	E	E
3.2.3.1 (text < 600 char.)	N	N	E	N
3.2.3.2 (tabular < 300 char.)	E	E	E	E
3.2.3.3 (graphic < 100 char.)	U	E	E	N
3.3 Data Aggregation	E	E	U	E
3.3.1 high (summary display)	E	E	U	N
3.3.2 moderate (grouped items)	E	E	U	U
3.3.3 low (individual items)	E	E	U	E
3.4 Data Coding	U	E	U	N
3.4.1 dimensionality	U	E	U	-
3.4.1.1 many relevant variables	N	N	U	-
3.4.1.2 few relevant variables	U	E	U	-
3.4.2 categories ("alphabet")	U	E	U	-
3.4.2.1 many (> 20) (alphanumeric)	N	N	U	-
3.4.2.2 moderate (8-20) (symbol)	N	N	U	-
3.4.2.3 few (3-7) (color)	U	E	U	-

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
3.4.2.4 just two (size, brightness, etc.)	U	E	U	-
3.4.3 criticality	N	E	N	-
3.4.3.1 high (redundant coding)	-	E	-	-
3.4.3.2 moderate (separate coding)	-	E	-	-
3.5 Display Partitioning	E	E	U	N
3.5.1 fixed windows	E	E	U	-
3.5.2 variable windows	N	N	U	-
3.5.2.1 automatic	-	-	N	-
3.5.2.2 by request	-	-	U	-
3.5.3 multiple displays	N	U	N	-
3.6 Display Selection	E	E	E	E
3.6.1 by display name	E	E	U	N
3.6.2 by data subset name ("category", "page", etc.)	E	U	U	N
3.6.3 by data item	E	E	E	N
3.6.4 by data characteristics ("selective retrieval")	U	N	E	E
3.6.4.1 automatic	U	-	N	E
3.6.4.2 by request	U	-	E	N
3.7 Data Coverage	E	U	E	N
3.7.1 displacement	E	N	E	-
3.7.1.1 page (text or tabular)	E	-	E	-
3.7.1.1.1 forward	E	-	E	-
3.7.1.1.2 back	E	-	E	-
3.7.1.2 scroll (text)	N	-	N	-
3.7.1.2.1 up	-	-	-	-
3.7.1.2.2 down	-	-	-	-
3.7.1.3 offset (graphic)	U	-	U	-
3.7.1.4 return/recenter	U	-	U	-
3.7.2 expansion	U	U	U	-
3.7.2.1 incremental	U	U	U	-
3.7.2.2 continuous	N	N	U	-
3.7.2.2.1 zoom in	-	-	U	-
3.7.2.2.2 zoom out	-	-	U	-
3.7.2.3 return/normalize	U	U	U	-

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
3.7.3 other reformatting	N	U	U	-
3.8 Display Update	N	E	U	N
3.8.1 automatic	-	E	N	-
3.8.2 by request	-	N	U	-
3.8.3 rate	-	E	U	-
3.8.3.1 normal (event driven)	-	E	N	-
3.8.3.2 fast (time compression)	-	U	U	-
3.8.3.3 slow	-	U	N	-
3.8.3.4 freeze ("stop action")	-	U	N	-
3.9 Data Deletion	N	U	U	E
3.9.1 automatic	-	N	N	N
3.9.2 by request	-	U	U	E
3.9.2.1 all ("CLEAR")	-	N	N	N
3.9.2.2 data category (selective erasure)	-	U	U	E
3.9.2.3 data item	-	N	U	E
4.0 SEQUENCE CONTROL				
4.1 Transaction Selection	E	E	E	N
4.1.1 general "OPTIONS"	E	E	U	-
4.1.2 contingent (step-specific) options	E	E	U	-
4.1.2.1 automatic	E	E	N	-
4.1.2.2 by request	E	E	U	-
4.1.3 linked commands ("macros")	N	U	E	-
4.2 Interrupt	E	E	E	E
4.2.1 "BACKUP"	N	U	N	E
4.2.2 "CANCEL"	E	U	N	E
4.2.3 "RESTART"	U	N	N	E
4.2.4 abort	E	E	U	N
4.2.5 "END"	U	N	E	E
4.3 Context Definition	E	E	U	E
4.3.1 automatic	N	N	N	E

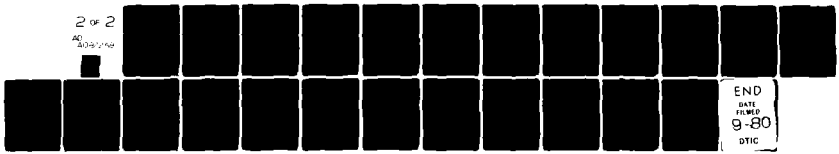
Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
4.3.2 by request	E	E	U	N
4.3.2.1 user command	E	E	U	-
4.3.2.2 data category	E	U	U	-
4.3.2.3 data item	E	E	U	-
4.4 Error Management	E	E	E	E
4.4.1 command validation	N	U	E	N
4.4.2 explicit "ENTER"	N	N	E	E
4.4.3 "CONFIRM" protection	U	E	E	E
4.4.4 direct error correction	N	N	E	E
4.5 Alarms	N	E	U	N
4.5.1 alarm definition	-	E	U	-
4.5.1.1 predefined	-	E	N	-
4.5.1.2 user defined	-	E	U	-
4.5.1.2.1 categories	-	E	U	-
4.5.1.2.2 parameters	-	E	U	-
4.5.2 alarm acknowledgment	-	E	U	-
4.5.2.1 automatic	-	U	U	-
4.5.2.1.1 routine ("timeout")	-	U	N	-
4.5.2.1.2 implicit action ("seen")	-	U	U	-
4.5.2.1.3 other	-	N	N	-
4.5.2.2 user action	-	E	U	-
4.5.2.2.1 predefined response	-	E	N	-
4.5.2.2.2 user defined response	-	U	U	-
4.5.2.2.3 override	-	N	N	-
5.0 USER GUIDANCE				
5.1 Status Information	U	E	E	U
5.1.1 operability	U	E	E	U
5.1.1.1 local work station	U	U	E	U
5.1.1.1.1 equipment	U	U	U	U
5.1.1.1.2 data files	U	U	E	U
5.1.1.1.3 functions	U	U	U	N
5.1.1.2 system	N	E	N	U
5.1.1.3 external	N	E	N	N
5.1.2 current users	N	N	U	N

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
5.1.3 current load (response time)	N	N	U	N
5.1.4 timeliness	U	E	U	U
5.1.4.1 continuous	N	E	N	N
5.1.4.2 periodic	N	N	N	N
5.1.4.3 by request	U	N	U	U
5.1.4.4 date/time	U	E	U	N
5.1.5 alarm signals	N	E	U	N
5.1.5.1 categories	-	E	U	-
5.1.5.2 parameters	-	E	U	-
5.2 Routine Feedback	E	E	E	E
5.2.1 input	E	E	E	E
5.2.1.1 data entry	E	N	E	E
5.2.1.2 data change	E	N	E	E
5.2.1.3 data deletion	N	E	E	E
5.2.2 output	E	E	E	N
5.2.2.1 requested data displayed	E	E	E	-
5.2.2.2 exceeds display capacity	E	N	E	-
5.2.2.2.1 partial display	E	N	E	-
5.2.2.2.2 not displayed	N	N	N	-
5.2.2.3 data not available	E	N	E	-
5.2.3 sequence control	E	E	E	N
5.2.3.1 requested transaction	E	E	E	-
5.2.3.2 changed context	E	E	E	-
5.2.3.3 other	N	U	E	-
5.3 Error Feedback	E	E	E	E
5.3.1 error type	E	E	E	E
5.3.2 correction procedure	E	E	U	N
5.3.3 alert signals	E	E	U	E
5.3.4 cursor position	E	E	U	N
5.4 Instructional Aids	E	E	E	E
5.4.1 automatic	E	E	U	E
5.4.1.1 fixed guidance messages	E	E	U	E
5.4.1.2 contingent on input	E	E	U	E
5.4.1.2.1 command selection	E	E	U	N
5.4.1.2.2 context change	N	U	U	N
5.4.1.2.3 data entry	U	N	U	E

AD-A087 258 MITRE CORP BEDFORD MA F/G 17/2
REQUIREMENTS DEFINITION AND DESIGN GUIDELINES FOR MAN-MACHINE I--ETC(U)
JUN 80 S L SMITH F19628-80-C-0001
UNCLASSIFIED M80-10 ESD-TR-80-122 NL



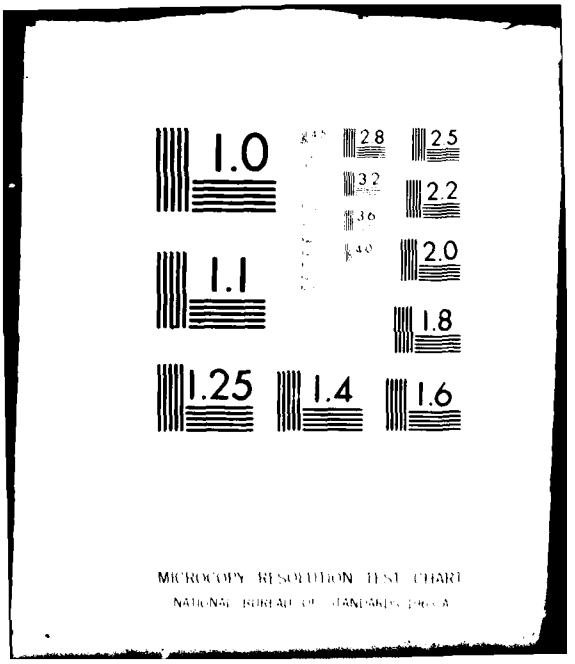

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

Table B-1 (continued)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
5.4.1.3 command aiding	N	U	U	N
5.4.1.3.1 branching options	-	U	U	-
5.4.1.3.2 disambiguation	-	N	U	-
5.4.1.3.3 other	-	N	U	-
5.4.1.4 cursor position	U	E	U	N
5.4.2 by request	E	E	E	N
5.4.2.1 command index	E	E	E	-
5.4.2.2 data index	N	E	E	-
5.4.2.3 "HELP", "EXPLAIN"	U	U	E	-
5.4.2.4 on-job training	U	N	U	-
5.4.2.5 other (simulation, etc.)	N	U	U	-
5.4.3 instructional modes	E	E	E	E
5.4.3.1 novice users	U	N	E	E
5.4.3.2 transitional users	N	N	U	N
5.4.3.2.1 by time of use	-	-	N	-
5.4.3.2.2 by demonstrated skill	-	-	U	-
5.4.3.2.3 other	-	-	U	-
5.4.3.3 expert users	U	U	U	N
5.4.3.4 mixed user skills	E	E	E	N
6.0 DATA TRANSMISSION/COMMUNICATION				
6.1 Data Transfer	E	E	E	E
6.1.1 to files	E	E	E	E
6.1.1.1 own	E	E	E	E
6.1.1.2 other users	U	N	U	N
6.1.2 to other terminals	N	N	U	N
6.1.3 to printer	U	U	E	E
6.1.4 external	E	E	N	U
6.2 Data Type	E	E	E	E
6.2.1 text	E	E	E	N
6.2.1.1 formatted	E	E	E	-
6.2.1.1.1 messages	E	E	E	-
6.2.1.1.2 documents	N	N	E	-
6.2.1.2 unformatted	N	N	E	-
6.2.2 tabular	E	E	E	E
6.2.3 graphic	N	N	U	N

Table B-1 (concluded)

MMI Capability	Characteristic Tasks MS	OM	IA	DE
6.2.4 alarm/alert signals	N	E	N	N
6.3 Transmission Control	E	E	E	E
6.3.1 data source	E	E	E	E
6.3.1.1 from display	U	N	E	N
6.3.1.2 from files	E	E	E	E
6.3.2 data specification	E	E	E	N
6.3.2.1 by display name	N	N	E	-
6.3.2.1.1 all	-	-	E	-
6.3.2.1.2 designated part	-	-	U	-
6.3.2.2 by data name	E	E	N	-
6.3.2.2.1 category	E	E	-	-
6.3.2.2.2 item	U	E	-	-
6.3.3 initiation	E	E	E	E
6.3.3.1 automatic	N	E	N	E
6.3.3.1.1 continuous	-	N	-	N
6.3.3.1.2 periodic	-	N	-	N
6.3.3.1.3 contingent on transaction	-	E	-	E
6.3.3.2 by request	E	E	E	N
6.3.4 feedback	E	E	U	N
6.3.4.1 transmission	E	E	U	-
6.3.4.1.1 initiated	N	E	N	-
6.3.4.1.2 confirmed	E	E	U	-
6.3.4.1.3 failed	E	E	U	-
6.3.4.2 message received	N	E	U	-
6.3.4.2.1 priority	-	E	U	-
6.3.4.2.2 routine	-	E	U	-

APPENDIX C

SAMPLE DESIGN GUIDELINES FOR DATA ENTRY

In Section 3 of this report, it is recommended that guidelines be prepared pertinent to the design of the man-machine interface for C3 systems. In this appendix, in Table C-1, a sample set of 72 guidelines is provided for design of functional capabilities relating to data entry. These MMI functions are those that have been given codes 2.x in the requirements matrix presented in Appendix B.

These design guidelines for data entry are not invented here for the first time. Most of them have been taken, albeit reordered and sometimes reworded, from more comprehensive sets of guidelines proposed by Engel and Granda (1975), and by Pew and Rollins (1975). Those prior authors deserve credit for setting down first what many people imagined they knew already but had not explicitly formulated.

These guidelines represent just a partial, tentative sample of what might be derived from published recommendations. Similar guidelines could be listed for other functional areas of MMI design. Even the data entry functions chosen for illustration here are not completely represented by these sample guidelines. There are, for example, no guidelines here that are specifically pertinent to graphical data entry. Presumably those could be derived from other reference sources.

Like the requirements matrix, the list of design guidelines will grow larger through cumulative experience with its use in trial applications. In this present listing, however, something of the variety of possible "rules" is evident. Some rules are simple, almost too simple to need stating. Others are complex. Some rules seem to apply generally to data entry functions, whereas others are directed toward specific sub-functions. The guidelines listed here are grouped in relation to the primary sub-functions of data entry, as coded in the requirements matrix.

What is missing from this sample list? Aside from graphics, many other specific sub-functions are poorly represented. There are undoubtedly many special-purpose rules still to be stated, which pertain to particular data entry applications.

Conversely, there may be some rules so elementary, so generally accepted, that no one thinks of them and they are never explicitly

stated. That is a mistake. Even the simplest rules deserve some sort of acknowledgment, as a reminder to the designer.

An example is the recommendation that a displayed cursor should not obscure another symbol whose position it may mark (see rule 2.1-2 in Table C-1). This rule seems self-evident, yet it is missing from published guidelines. Once in a while, however, a display will be designed in such a way that when its cursor marks another character that character is temporarily erased. Anyone who has to work with such a display will be reminded of the need for this "missing" rule.

One obvious deficiency in the guidelines listed here is the infrequent inclusion of examples. For the MMI designer, examples might help to clarify the intent and significance of the rules. If lists such as this are to be published as design standards, they should be written to include more examples.

Some guidelines relating to the provision of routine feedback for data entry are included here, although they relate also to functional capabilities for User Guidance (code 5.2 in the requirements matrix). Not included here are guidelines for other User Guidance functions relating to data entry, such as feedback for error correction (code 5.3) and instructional aids for data entry (subfunctions under code 5.4). Also not included here are Sequence Control functions associated with data entry, including interrupt functions (code 4.2) and error management (code 4.4).

Another deficiency of this sample list is its almost exclusive concern with electronic visual displays as the input/output medium for man-machine interaction. A few references to auditory displays are included, but more are needed. Additional guidelines would also be needed for MMI designs employing a teletype (or reactive typewriter) rather than electronic displays.

The dependence of design on its means of implementation is acknowledged in the contingent wording of some of the guidelines listed here. For the most part, however, the list includes functional rules rather than hardware specification. Such functional guidelines could be amplified to include a more explicit recognition of hardware features if that were appropriate in system acquisition.

Certainly there are hardware features implied in these guidelines, including function keys for ENTER, CONFIRM, BACKUP, CANCEL, DITTO, PRINT, possibly DEFAULT and RESTART, as well as various tab keys for controlling cursor position, for MMI designs

involving form-filling dialogues. It is not clear whether it would prove desirable in system acquisition to state the hardware implications of these guidelines more specifically. It may be better to leave that to the designer.

One significant weakness in current design guidelines is that they are based on opinion, judgment, accumulated wisdom, rather than on quantitative performance measures. Thus the designer is given good advice but not told the consequences of ignoring it. Until research on man-machine interaction catches up with application, which may take a long time, this weakness cannot be remedied. For the present, it can be argued that good advice, even if ignored, is probably better than no advice at all.

If design guidelines are based on judgment, then they are potentially controversial. Thus any proposed guidelines should be examined through extended debate and in broad-ranging application before they can be adopted as a general design standard. Some rules that sound good for the general case may fail in a specific application. Such exceptions should be noted as they are discovered. Meanwhile, readers of this report are invited to propose improvements and additions to the sample guidelines presented here.

As any initial list of guidelines is gradually amended and extended, the eventual result may become quite a large design handbook. For any particular system application the large total set of guidelines would need tailoring to the particular dialogue type(s) selected for use and in accord with the MMI requirements matrix for that system. Procedures for tailoring design guidelines must be worked out in practice. As suggested in the text (see page 18), computer-based aids could be helpful in this regard. The result of such tailoring, in combination with a detailed requirements definition, should provide an effective functional specification for the man-machine interface.

Table C-1

Design Guidelines for Data Entry Functions

Code*	
2.0	DATA ENTRY/INPUT
- 1	Where data entry is a significant task function, it should be accomplished via the operator's primary display. (For example, entry via typewriter is acceptable only if the typewriter itself, under computer control, is the primary display medium.)
- 2	Data entry transactions, and associated displays, should be designed so that the operator can use one mode of entry as long as possible before having to switch to another (e.g., switching from lightpen to keyboard input).
- 3	Except for passwords and other secure entries, keyed inputs should always appear in the display.
- 4	Keyed data entry and change on an electronic display should generally be accomplished by direct character replacement, in which keyed inputs replace underscores or previous entries (including default values) in defined data fields.
- 5	Wherever possible, data entry should be self-paced, depending upon the operator's needs, attention span and time available, rather than computer processing or external events. (Where self-pacing does not seem feasible, the general approach to task allocation and MMI design should be reconsidered.)
- 6	Using a form-filling dialogue, entry of logically grouped items should be accomplished by a single, explicit action at the end, rather than requiring separate entry of each item.

*Refers to coding scheme of functional capabilities adopted for the MMI requirements matrix in Appendix B (see Table B-1).

Table C-1 (continued)

- 7 Where multiple items are entered as a single transaction, the operator should be allowed to "back up" and change any item before taking the final entry action.

2.1 Position Designation (Cursor Control)

- 1 Position designation on an electronic display should be accomplished by means of a movable cursor with distinctive visual features (shape, blink, etc.). (However, where position designation involves only selection among displayed alternatives, then some form of highlighting selected items might be used instead of a separately displayed cursor.)

- 2 The cursor should be designed so that it does not obscure any other character which may be displayed in the position designated by the cursor.

- 3 Where fine accuracy of positioning is required (as in some forms of graphic interaction) the displayed cursor should include a point designation feature.

- 4 Actual entry ("activation") of a designated position should be accomplished by an explicit operator action distinct from cursor placement.

- 5 For arbitrary position designation, the cursor control should permit both fast movement and accurate placement. Rough positioning should take no more than 0.5 seconds for a displacement of 20-30 cm on the display. Fine positioning may require incremental stepping of the cursor, or a control device incorporating a large control/display ratio for small displacements, or a selectable vernier mode of control use.

- 6 In position designation, the displayed cursor should be stable, i.e., should remain where it is placed until moved by the operator (or computer) to another position.

Table C-1 (continued)

- 7 Continuous position designation, such as used for input of track data, should be accomplished by continuously operable controls (e.g., thumb wheel for one dimension, joystick for two dimensions) rather than incremental, discrete key actions.

- 8 When position designation is the sole or prime means of data entry, as in selection of displayed alternatives, cursor placement should be accomplished by a direct-pointing device (e.g., lightpen) rather than by incremental stepping or slewing controls (keys, joystick, etc.).

- 9 In selection of displayed alternatives, the acceptable area for cursor placement should be made as large as possible, including at least the area of the displayed label plus a half-character distance around the label.

-10 Where position designation is combined with keyed data entry, cursor movement should be controlled at the keyboard (by function keys, joystick, "cat", etc.) rather than by a separately manipulated device (lightpen, "mouse", etc.).

-11 On initial appearance of a data entry display the cursor should be placed automatically at the first character position of the first input field.

-12 Display formats for data input should be designed so as to minimize operator actions required for cursor movement from one entry field to the next.

-13 Sequential cursor positioning in predefined areas, such as displayed data entry fields, should be accomplished by programmable tab keys.

-14 Areas of a display not needed for data entry (such as labels and blank spaces) should be made inaccessible to the operator, under computer control, so that the cursor does not have to be stepped through blank areas nor are they sensitive to pointing actions. (Mechanical overlays on the display should not be used for this purpose.)

Table C-1 (continued)

-15 Operator action confirming entry of multiple data items should result in input of all items, regardless of where the cursor is placed on the display.

2.2 Direction Designation

- 1 Where accurate designation of direction is required, some "analog" means of entry should be provided, such as vector rotation on the display, and/or a suitably designed rotary switch (see Smith, 1962). Where only approximate direction designation is required, for just eight cardinal points, keyed entry can be used.

2.3 Data Type

- 1 Ideally, the length of individual data entries should not exceed 5-7 characters, except for textual material. Longer items (such as the 17-character transportation control number for air cargo) exceed the operator's memory span, inducing errors in both data entry and review.

- 2 Where longer items must be entered, the item should be partitioned into shorter symbol groups for both entry and display (e.g., a 10-digit telephone number entered as three groups: ___ ___ ____).

- 3 Where portions of a long item are highly familiar or rendundant, ideally those should be entered last, in order to reduce the load on the operator's short-term memory (but not if this sequence would violate a functional requirement, such as initial keying of area code in telephone numbers).

- 4 Minimize keying in data entry by abbreviation of lengthy inputs where that can be done without ambiguity, providing data validation routines and operator interrogation as necessary to resolve any ambiguity which may arise.

- 5 Where abbreviated codes are used to shorten data entry, code values should be designed to be as distinctive as possible in order to avoid potentially

Table C-1 (continued)

confusing similarity (e.g., BOS vs. LAX is good; LAS vs. LAX is bad).

- 6 Where alphabetic data entry is required, restricted alphabetic sets should not be used. Software might be provided to interrogate the operator to resolve any input ambiguities resulting from hardware limitations (see Smith and Goodwin, 1971).

- 7 Special characters used in data entry (, * = / etc.), particularly if used frequently, should be chosen insofar as possible so that the keyboard operator will not have to shift from one case to another. (Conversely, keyboard designers should put frequently used special characters where they can be most easily keyed.)

- 8 Entry of leading zeros should be optional for general numeric input, though it may be required occasionally for special cases (e.g., entry of serial numbers or similar identifiers.)

- 9 For input of tabular data, where vertical repetition of entries is frequent the operator should be provided a "ditto" key to speed entry of duplicative data.

2.4 Entry Formats

- 1 Wherever possible, multiple data items should be entered without the need for special separators or delimiters, either by keying into predefined entry fields or by including simple spaces between sequentially keyed items. Where a delimiter must be used, adopt a standard character; slash (/) is preferred.

- 2 For all dialogue types involving prompting, data entries should be prompted explicitly by means of displayed labels for data entry fields, and/or associated user guidance messages.

Table C-1 (continued)

- 3 For operator-initiated (command language) dialogues, a means of entry stacking should be available so that an experienced operator can shortcut prompting sequences.

- 4 For data entries involving selection among displayed alternatives, implicit prompting can be provided by marking (brightening, inverse video, etc.) the selected alternative(s).

- 5 Implicit cues should be provided in form-filling dialogues to supplement explicit labels. Special characters (e.g., underscores) should be used to delineate each entry field.

- 6 Field delineation cues should distinguish required (dashed underscore) from optional (dotted underscore) entries, and should indicate the maximum acceptable length of the entry. (Similar implicit cues can be provided when data entry is prompted by auditory displays. See Smith and Goodwin, 1970.)

- 7 Where item length is variable, the operator should not have to justify an entry either right or left, and should not have to remove any unused underscores.

- 8 Where multiple items (especially those of variable length) will be entered by a skilled touch typist, each entry field should end with an extra (blank) character space. This will permit consistent use of tab keying to move from one field to the next. An auditory signal should be provided to alert the operator if an entry is keyed into a blank space.

- 9 Where entry fields are distributed across a display, a consistent format should be adopted for relating labels to delineated entry areas. For example, the label might always be to the left of the field; or the label might always be immediately above and left-justified with the beginning of the field. Such consistent practice will help the operator distinguish labels from data in distributed displays.

Table C-1 (continued)

-10 Where tabular formats are used for data entry, column labels should be left-justified with the leftmost position beginning column entries, especially when columns vary in width.

-11 Numeric entries (e.g., dollars and cents), although entered as left-justified, should be automatically justified with respect to a fixed decimal point if a display of these data is regenerated for review by the operator.

-12 Labels for data entry fields should be distinctively worded, so that they will not be readily confused with data entries, labeled control options, guidance messages, or other displayed material.

-13 In labeling data entry fields, only approved terms, codes and/or abbreviations should be used. Do not create new jargon. If in doubt, pretest all proposed wording with a sample of qualified users.

-14 Labels for entry fields may incorporate additional cueing of data formats where that seems helpful, for example, DATE (M/D/Y): __ __ __.

-15 Where a dimensional unit ($, mph, km, etc.) is consistently associated with a particular data field, it should be displayed as part of the fixed label rather than entered by the operator. Where alternative dimensional descriptors are acceptable, then space should be provided in the data field for operator entry of a unit designator.

-16 Where data entry displays are crowded, auxiliary coding should be adopted to distinguish labels from data. The recommended standard is to display fixed, familiar labels in dim characters, with data entries bright.

-17 Display formats for data entry should be identical or compatible with formats used for display output, scanning and review of the same data items. Where a display format optimized for data entry seems unsuited

Table C-1 (continued)

for data review, or vice versa, some compromise format should be designed taking into account the relative functional importance of data entry and review in the operator's task.

-18 Where data entry involves transcription from source documents, in a form-filling dialogue displayed forms should match (or be compatible with) paper forms. In a question-and-answer dialogue, the sequence of entry should match the data sequence in source documents. (Where paper forms are not optimal for data entry, consider revising the layout of the paper form.)

-19 Where data entries must follow an arbitrary sequence of external inputs (e.g., keying telephoned reservation data), some form of command language dialogue should be used to identify each item as it is entered, so that the operator does not have to remember and re-order items.

-20 If no source documents or external inputs are involved, the ordering of multiple-item data entries should follow the logical sequence in which the operator can be expected to think of them. Alternatively, data entry can sometimes be made more efficient by placing all required fields before any optional fields.

2.5 Data Validation

- 1 Automatic data validation software should be incorporated to check any entry whose input and/or correct format or content is required for subsequent data processing. Do not rely on the operator always to make correct inputs.

- 2 Where critical data entries have not been input, but can be deferred, data validation software should signal that to the operator, permitting either immediate or delayed input of missing items.

- 3 Where deferred entry of a required data item is necessary, the operator should have to enter a special

Table C-1 (continued)

symbol in the data field to indicate this item has been temporarily omitted rather than ignored.

- 4 In a repetitive data entry task, data validation for one transaction should be completed, and the operator allowed to correct errors, before another transaction can begin. (This is particularly important when the operator is transcribing data from source documents, so that any detected input errors can be corrected while the relevant document is still at hand.)

- 5 Item by item data validation within a multiple-entry transaction may be helpful to a novice user, but if this capability is provided it should be only as a selectable option. It will tend to interrupt and slow a skilled operator.

- 6 Where useful default values for data entry cannot be predicted by system designers, which is often the case, the operator (or perhaps some authorized supervisor) should be provided a special transaction to define, change or remove default values for each data entry field.

- 7 Currently defined default values should be displayed automatically in their appropriate data fields with initiation of a data entry transaction. The operator should not be expected to remember them.

- 8 Operator acceptance of a default value should be accomplished by simple means, such as by a single confirming key input or by tabbing past the default field. (Similar techniques should be used in tasks involving operator review of stored data.)

- 9 In any particular data input transaction the operator should be able to replace any default value with a different entry, without changing current default definition.

Table C-1 (continued)

2.6 Data Processing

- 1 The user should not be required to enter redundant data, already known to the computer, except as needed for resolving ambiguous entries, security (user identification), verification of stored data (better handled by review rather than re-entry) or operator training. (For example, the operator should not have to enter both an item name and identification code when either one defines the other.)

- 2 Data entries made in one transaction should be remembered by the system when relevant to another transaction, and displayed for review if appropriate. The operator should not have to enter those data again.

- 3 Wherever needed, automatic computation of derived data should be provided, so that the operator does not have to calculate and enter any number that can be derived from data already entered in the system.

- 4 Wherever needed, automatic cross-file update should be provided, so that the operator does not have to enter the same data twice (e.g., assignment of aircraft to a mission should automatically indicate that commitment in squadron status files as well as in a mission assignment file).

5.2 Routine Feedback

- 1 In tasks where data input is usually a single, occasional transaction, successful entry should be signaled by a confirmation message without removing any visual display of the entered data. (This follows a general principle of sequence control that the operator should leave one transaction and choose the next by explicit command.)

- 2 In tasks where data input is usually repetitive, in a continuing sequence of transactions, successful entry should be signaled by regeneration of the data entry display, automatically removing the just entered data

Table C-1 (concluded)

in preparation for the next entry. (This represents an exception to the general principle of sequence control by explicit operator choice, in the interest of efficiency.)

- 3 Where an entry will cause any extensive change in stored data, procedures and/or system operation, and particularly if that change cannot be easily reversed, the operator should be notified and required to confirm the entry action before it is implemented.

- 4 In an extended data entry transaction, wherever possible a cumulative record of operator inputs should be displayed if relevant to the current input. (In a multi-page data entry display, for example, do not rely on the operator to remember data accurately from one page to the next.)

- 5 Keys and other input devices not needed for data entry, particularly those with disruptive consequences, should be temporarily disabled under computer control, during data entry transactions. (Mechanical overlays manipulated by the operator should be used only as a last resort.)

- 6 Where computer processing of just-entered data will result in a noticeable delay, an acknowledging message should appear, and the keyboard should be automatically locked until the operator can begin a new transaction. Keyboard lock should be accompanied by disappearance of the cursor from the display and (especially if infrequent) by some more specific indicator such as an auditory signal.

- 7 Where the operator requires a printed record of data entries, a print request should not change the contents of the display, except for the addition of a confirmation message if printing is accomplished remotely.

APPENDIX D

DOCUMENTING MMI DESIGN

In Section 4 of this report it is argued that definition of functional requirements and establishment of guidelines, although worthy activities, are not sufficient to ensure effective MMI design. It will be necessary to document a proposed design for review and coordination by the various groups of people responsible for system development and operation.

Design documentation should certainly provide an account of requirements met and guidelines followed, but should include other material as well. Pew and Rollins (1975) recommend five components for design documentation. With some changes of terminology, these five components can be summarized as follows.

1. Task Flow Chart. This breaks down a generally stated operator task into a logical series of decisions to be made and information handling activities to be performed. Some of these activities will involve interaction with a computer subsystem, and some may not. Task analysis at this level may be provided in a system functional specification, but more often must be developed in the first stage of system design.

2. Transaction Flow Chart. This breaks down those activities involving computer use into a series of discrete transactions. The proposed man-machine dialogue is outlined step by step, or as is sometimes said for dialogues involving visual displays, frame by frame. The transaction flow chart indicates choices the operator can make in sequence control, contingent branching as a result of operator input, data entries required, outputs generated, etc.

3. Display (Frame) Format. Working from the transaction flow chart, each step or display frame must be specified in detail. The most effective way to do this is to design a facsimile display showing the exact wording, spacing, etc., that will be used. Here is where MMI guidelines could be particularly helpful to the designer.

4. Input Data Definition. For transactions involving data entry, it will be necessary to provide a definition for each item (or "variable") to be input. This definition will involve specifying the range of acceptable values for each entry where data validation is required, the consequences of data entry in terms of required computer processing, and the implications (if any) for subsequent sequence control and for user guidance.

5. User Guidance. For any transaction, but particularly for transactions involving data entry, the designer should specify the feedback that will result from any possible operator action, e.g., alarm or alert signals, guidance messages, other display changes. Often this aspect of MMI design is conceived more narrowly, in terms only of the specification of error messages. It will help MMI design, however, if the designer can adopt a broader perspective. Perhaps the operator should be given a message requesting confirmation of a doubtful entry, i.e., an entry that is so unlikely that it is probably though not certainly an error. Perhaps the operator should be requested to enter additional data to amplify an entry just made. Or perhaps the operator should simply be given a message suggesting the next step in a transaction sequence. There are many possible aspects of user guidance that cannot fairly be termed error messages.

In their report, Pew and Rollins suggest that special forms be adopted for documentation of these various aspects of MMI design. Examples of their forms are reproduced here, with permission, in the remaining pages of this appendix, to give an idea of what may be needed for documentation purposes. The task chosen for illustration has to do with commodity loan repayment, for an information system being considered at that time (1975) by the U.S. Department of Agriculture. Of interest here, however, is the general approach used rather than the particular task being documented. Other forms of documentation might work as well, or better, as long as a similar degree of detail is achieved.

Pew and Rollins make the additional recommendation that once MMI design has been documented it should be subject to configuration management control. That is to say, during software implementation changes to a documented MMI design should be made only after review and authorization. Such control will help ensure design

consistency, so that different people can implement different parts of the MMI with reasonable confidence that the parts will fit together as a coherent whole. That seems a good idea.

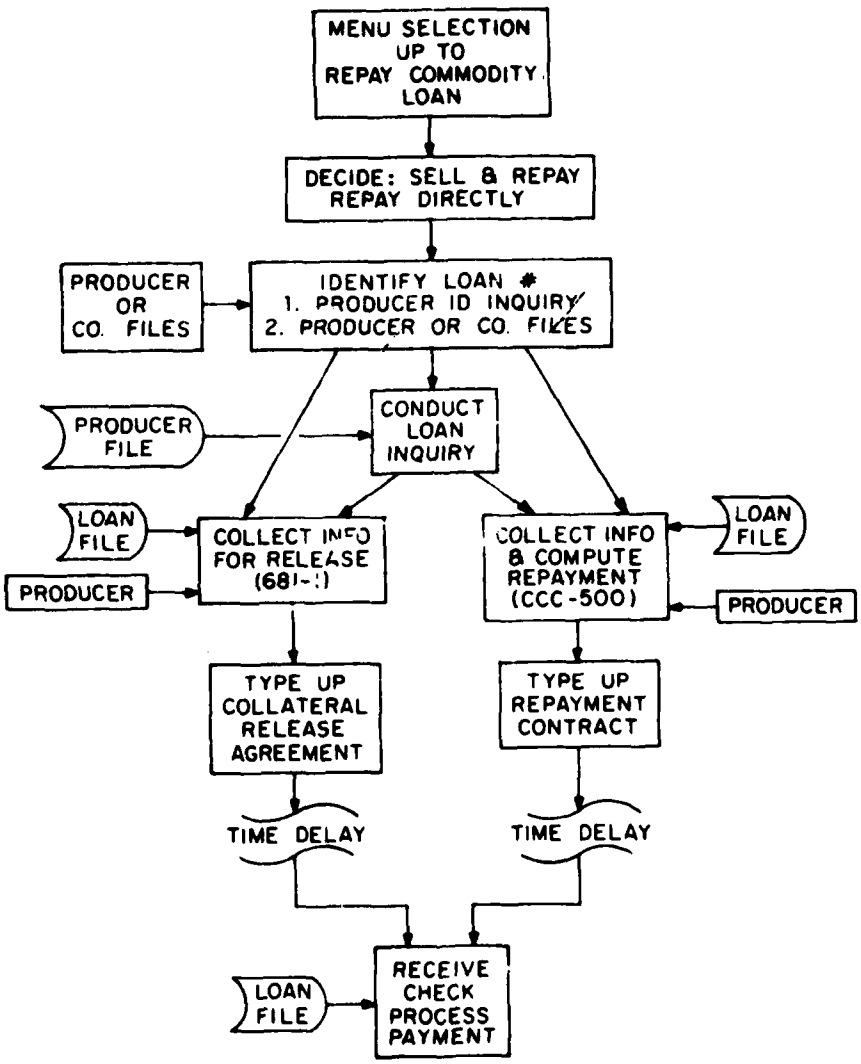
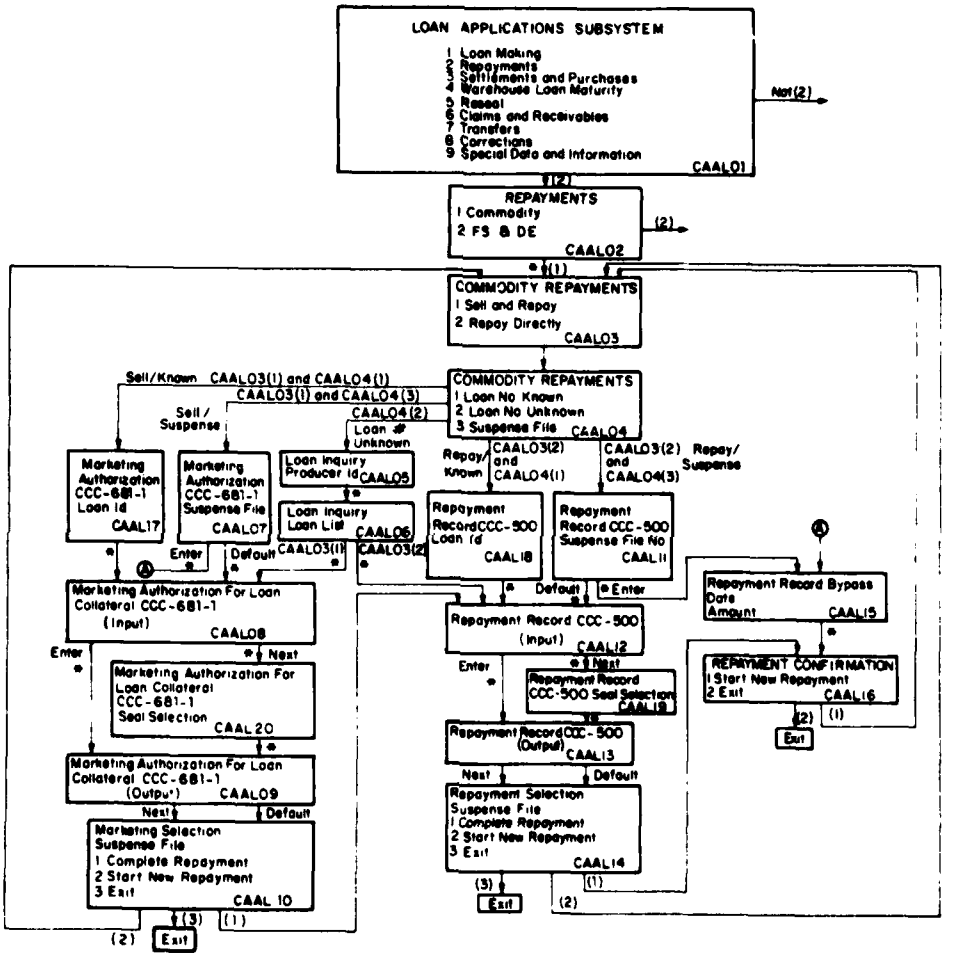


Figure D-1. Sample Task Flow Chart



*indicates a point at which access to central computer data base is required

Figure D-2. Sample Transaction Flow Chart

DIALOG FRAME SPECIFICATION

FRAME NO.	PAGE	DATE	FRAME TYPE
CAAL18	1 OF 1	06/12/75	INPUT ☒ OUTPUT ☐ MENU ☐

SUBSYSTEM NAME: LOANS

LINE NO	CHARACTER POSITION
1	REPAYMENT RECORD CCC-500,P1
2	
3	PRODUCER ID
4	ST. AND CO. CODES
5	LOAN NO.
6	LOAN TYPE
7	COMMODITY
8	CROP YEAR
9	
10	
11	PRESS "ENTER" TO RETRIEVE LOAN RECORD.
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	

COMMENTS: COLLECT INFORMATION ABOUT THE LOAN

DESIGNED BY: AMR

Figure D-3. Sample Form for Display Format Specification

VARIABLE DEFINITION

FRAME NO: CAAL18 | PAGE: 1 OF 1 | DATE: 06/12/75

SUBSYSTEM NAME: LOANS

VAR NO	LINE NO	STARTING CHAR POSITION	VARIABLE EDIT CRITERIA: NO OF CHAR	TYPE: ALL ALPHA	TYPE: ALL NUMERIC	TYPE: MIXED	NUMERIC RANGE OR PERMITTED CHARACTER	MANDATORY: Y	MANDATORY: N	DEFAULT VALUE	HELP MESSAGE	COMMENTS
1	3	14	10		X			X				PRODUCER ID
2	4	20	6		X			X		LOCAL STATE LOCAL COUNTY	PRESS "DEFAULT" FOR LOCAL STATE AND COUNTY CODES	ST. & CO. CODES
3	5	11	4		X			X				LOAN NO.
4	6	12	1	X				X		MOST COMMON TYPE		LOAN TYPE
5	7	12	10	X				X				COMMODITY
6	8	12	4		X		1970-1975	X		1975		CROP YEAR

COMMENTS: LOAN IDENTIFICATION INFORMATION - REPAYMENT RECORD CCC-500

DESIGNED BY: AMR

Figure D-4. Sample Form for Input Data Definition

ERROR MESSAGE SPECIFICATION	FRAME NO. CAAL18	PAGE: 1 OF 1	DATE 06/12/75
	SUBSYSTEM NAME: LOANS		

VARIABLE IN ERROR 1 (PRODUCER ID)	VALIDATION THAT PRODUCED ERROR: SEARCH FOR PRODUCER ID

ERROR MESSAGE

SEARCH FOR PRODUCER 999-99-9999-9 FAILED. CHECK PRODUCER ID.

VARIABLE IN ERROR 3 (LOAN NUMBER)	VALIDATION THAT PRODUCED ERROR: LOAN NUMBER TO BE REPAID ASSIGNED TO INDICATED PRODUCER

ERROR MESSAGE

LOAN NUMBER NOT ASSIGNED TO THIS PRODUCER. CHECK LOAN NUMBER AND PRODUCER ID.

VARIABLE IN ERROR 5 (COMMOD-ITY)	VALIDATION THAT PRODUCED ERROR: VALIDITY OF COMMODITY CODE

ERROR MESSAGE

CORN LOAN NOT APPLICABLE IN 1975. CHECK COMMODITY OR DATE.

COMMENTS	DESIGNED BY
	AMR

Figure D-5. Sample Form for Documenting User Guidance Messages


MED
18